NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

THE UNIVERSITY OF MICHIGAN

COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
DEPARTMENT OF COMMUNICATION SCIENCES

Technical Report

Equivalences Between Probabilistic and Deterministic Sequential Machines

C. V. PAGE

Administered through:

April 1965

OFFICE OF RESEARCH ADMINISTRATION · ANN ARBOR

Requests for additional copies by agencies of the Department of Defense, their contractors, and other Government agencies should be directed to:

Defense Documentation Center (DDC)
Cameron Station
Alexandria, Virginia 22314

Department of Defense contractors must be established for DDC services or have their "need-to-know" certified by the cognizant military agency of their project or contract.

THE UNIVERSITY OF MICHIGAN

COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Department of Communication Sciences

Technical Report

EQUIVALENCES BETWEEN PROBABILISTIC AND DETERMINISTIC SEQUENTIAL MACHINES

C. V. Page

ORA Projects 03105, 06376, 06689, 06921

under contract with:

U. S. ARMY RESEARCH OFFICE (DURHAM)
GRANT NO. DA-ARO(D)-31-124-G558, PROJECT NO. 4049-M
DURHAM, NORTH CAROLINA

and

NATIONAL SCIENCE FOUNDATION
GRANT NO. GP-2539
WASHINGTON, D.C.

and

U. S. DEPARTMENT OF HEALTH, EDUCATION, AND WELFARE
PUBLIC HEALTH SERVICE, NATIONAL INSTITUTES OF HEALTH
GRANT NO. GM-12236-01
BETHESDA, MARYLAND

and

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
CONTRACT NO. Nonr-1224(21)
WASHINGTON, D.C.

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

May 1965

RESEARCH PROGRESS REPORT

Title: "Equivalences Between Probabilistic and Deterministic Sequential Machines," C. V. Page, The University of Michigan Technical Report 03105-37-T, April 1965; Nonr 1224(21), NR 049-114.

Background: The Logic of Computers Group of the Communication Sciences Department of The University of Michigan is investigating the application of logic and mathematics to the theory of the design of computing automata.

Condensed Report Contents: The concept of probabilistic sequential machines (PSM), a generalization of Rabin's concept of probabilistic automata, is defined. Such diverse devices as unreliable digital computers, slot machines, and chemical cells are presented as examples of PSM. Using the examples as motivation, various kinds of equivalences between machines are discussed. The fundamental question of when a PSM is equivalent in some sense to a deterministic machine, perhaps with random devices attached to output states, is considered. Finally various tests involving finitely many random variables are devised for each of the kinds of equivalences between PSM and for reduction, if possible, to deterministic machines. One of the tests is a further generalization of the Moore bound for deterministic machines than has previously appeared in the literature.

For Further Information: The complete report is available in the major Navy technical libraries and can be obtained from the Defense Documentation Center. A few copies are available for distribution by the author.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Concluded)

0. INTRODUCTION

The notion of behavioral equivalence is a fundamental part of the study of automata theory. Two definitions of behavioral equivalence occur in the literature for deterministic machines. One, due to Burks [5], calls two machines behaviorally equivalent if they define the same function from input strings to output strings. The other, part of Rabin-Scott automata theory, calls two machines behaviorally equivalent if they accept the same set of tapes. The two definitions can be shown to be the same for deterministic machines by recoding arbitrary output symbols into strings of zeros and ones. Both definitions have been generalized for probabilistic machines. However, for probabilistic machines the resulting generalizations are not equivalent.

This paper is concerned with certain kinds of equivalences between probabilistic machines. Two models will be discussed later in this section in order to gain insight into the main kinds of equivalences which will be studied. Of particular interest will be when a probabilistic sequential machine is equivalent in some sense to a finite deterministic machine.

0.1 THE CONCEPT OF PROBABILISTIC SEQUENTIAL MACHINE

By a probabilistic sequential machine is meant a system which satisfies one of the following two definitions:

Definition 0.1: A (Moore-type) probabilistic sequential machine A is a system $A = < n, I, S, \Sigma, A(0),\ldots,A(k-1), F, 0 >$ where

n: a natural number, the number of states

I: a n-dimensional stochastic vector, the initial state vector

S: set of state vectors = $\{S_1 = (1,0,\ldots,0),\ldots,S_n = (0,\ldots,0,1)\}$

Σ: alphabet set $\Sigma = \{0,1,2,\ldots,k-1\}$

$A(i)$: $i = 0,1,\ldots,k-1$ n x n switching matrix for input symbol i. $A(i)_{\ell m}$ is the probability of a transition from state ℓ to state m via symbol i.

F: output vector, a n-dimensional column vector whose entries are real numbers.

O: output function $O(S_i) = S_i \times F = F_i$: $S_i \in S$

Definition 0.2: A (Mealy-type) probabilistic sequential machine.

$A = < n, I, S, \Sigma, A(0),\ldots,A(k-1), W, P >$

where $n, I, S, \Sigma, A(0),\ldots,A(k-1)$ are as in 0.1 and where the output function P satisfies

$$P(S_i,j) = W_{ij} \quad S_i \in S,\ j \in \Sigma$$

It is an easy matter to show that Definition 0.1 and 0.2 are equivalent in the following sense: For every Moore-type probabilistic sequential machine there is a Mealy-type sequential machine whose output is the same random variable over each input and vice-versa. Consequently, we will be concerned only with the properties of Moore-type probabilistic sequential machines, which from now on will be called "sequential machines."

There seem to be many instances of systems like probabilistic sequential machines from other fields of study not generally thought to be automata theory. Braines and Svechinsky discuss a system like Definition 0.1 in their paper "Matrix Structure in Simulation of Learning" [1]. If one takes the cartesian product of machines of Definition 0.2, one gets the Markov processes with rewards and alternatives as studied in sequential decision theory as presented by Howard [2]. Matrix games as discussed by Thrall [3] can be considered as instances of Definition 0.1 in which I and F are strategy vectors and game matrix A(x) is defined by a string x. A simple correspondence shows that the noisy discrete channel of Shannon [8] is equivalent to the system of Definition 0.2. One would hope that someday probabilistic sequential machines could become a unifying concept, organizing and providing results for these diverse fields.

Probabilistic sequential machines are generalizations of the work of Rabin [4] for probabilistic automata. If one restricts I to elements of S and $F_i = 0$ or 1 for $i = 1,2,\ldots,n$ then Definition 0.1 defines probabilistic automata. Following Rabin, we remark that:

<u>Remark 1</u>: Let $x = i_1 \ldots i_r, i_j \in \Sigma, j = 1,\ldots,r.$
Then $A(x) = A(i_1)\ldots A(i_r)$ i.e. the switching matrix for a string x is found by multiplying the matrices for the symbols of x together in order.

0.2 MODELS OF PROBABILISTIC SEQUENTIAL MACHINES

We consider here two models, one of which can be considered probabilistic and one of which can be considered deterministic, although both fall within the framework of probabilistic sequential machines.

Example 0.1. Probabilistic internal operation: A slot-machine

A simple model of a probabilistic sequential machine is a slot-machine. The static position of the dials represents the present state of the machine.

Usually there are 20 different positions on the dial and 3 dials for a total of 8,000 states. The input consists of putting in a coin and pulling a lever, causing the machine to travel transiently through many states until it settles down in one state. An output is associated with each state. Nothing (which is associated with 0) comes out unless the dials all display the same object. In that case, some change tumbles out (which is associated with the corresponding real number) usually dependent only on the kind of object being displayed, i.e., the state of the machine. Such a machine whose output is controlled by its states is known as a "Moore machine" [7]. Each state can be associated with a number between 1 and 8,000, and the output for each state can be tabulated in a column vector or 8,000 x 1 matrix. In the formalism, this column vector will be called the "output vector" and designated by the symbol "F". The output for state i will be written as "F_i".

The enormous number of distinct ways the lever can be pulled are prevented from significantly influencing the outcome by spring loading. Hence for all practical purposes there is only one kind of transition law associated with pulling the lever. If the randomness of transition of the dials caused by variable factors like dust friction, humidity, heating and small vibrations does not change over long periods of time, the probability of a transition from any state of the dials to any other can be determined experimentally to any required precision. This situation is summarized formally in the assumption for probabilistic sequential machines that the transition probabilities are stationary. Symbolizing the usual lever play of the machine by L, the transition probabilities can be tabulated in a matrix

A(L), with the entry in the i'th row and j'th column (written $A(L)_{ij}$) being the probability of a transition from state i to state j via input L.

If there were no other permissible way to affect the rotation of the dials than by a pull of the lever, then the behavior of a slot-machine A could be described as a finite state Markov chain with rewards and transition matrix A(L). However, sudden small external shocks during the rotation of the dials can influence the state transitions of the machine. In order to model completely how such machines are played, we can consider a finite repeatable set of such non-standard inputs to the machine. For instance, one such input might be described as the application of a kick with a prescribed kinetic energy on a certain spot on the machine occurring 1/5 of a second after the lever is released. Symbolizing this manner of playing the machine by K, the transition matrix A(K) could be determined experimentally since the input is repeatable. A finite set of such repeatable inputs could be defined and their effects on the behavior of the machine ascertained.

To find out how strings of S and K inputs to the machine affect its operation, it is sufficient to multiply the matrices A(S) and A(K) together in the order specified by the string, e.g., if a string X is SKKSK then the transition matrix A(X) is the product $A(S) \cdot A(K) \cdot A(K) \cdot A(S) \cdot A(K)$.

Consider how the dials of the machine might be found initially. If the dials can be completely observed, the initial state of the machine is observable. In this case, in the formalism the initial state i is represented by a vector I (or a 1 x 8,000 matrix) with a 1 in the i'th component and zeros elsewhere. On the other hand, the dials may not be completely visible, and we may wish to specify the average behavior of a large number of machines run simultaneously, or we may wish to consider the average return from playing one machine only when it is left by other players on one of a set of preferred states. In any one of these cases, I can be a stochastic vector $(I_1,\ldots,I_{8,000})$ where I_i is the probability of being in state i at time t_0.

In the general case, the next state probabilities starting with an initial state vector I and an input string X are given by $I \cdot A(X)$. Hence the expected value of output of a machine A starting with initial state distribution I and output vector F after a string X of inputs has occurred is just

$$E_A(X) \quad = \quad I \cdot A(X) \cdot F$$

which is a bilinear form in I and F with form matrix $A(X)$. The variance in output and other higher moments can be defined analogously.

Example 0.2. Deterministic internal structure: Chemical production cell

Suppose a chemical tank A is divided into several isolated compartments $A_1,\ldots,A_n$ by partitions which are interconnected by an electronically controlled system of pumps and valves. Suppose that there is a finite set of controls $\Sigma = 0,1,\ldots,K-1$ and that for each control c a fixed fraction of the chemical in compartment A_i, v^c_{ij}, is pumped into compartment A_j. For all controls c in Σ, the full influence on redistribution of liquid in the tank can be described in a n x n matrix $A(c)$ with v^c_{ij} being $A(c)_{ij}$. Furthermore, suppose that the liquid being pumped between compartments is a catalyst which causes production of a desired end product in each compartment with a different efficiency, i.e., if the mass fraction of catalyst in A_i is P_i and F_i is the efficiency of A_i, then the output of end product is P_iF_i. Note that it is assumed that the output of the compartment depends linearly on the catalyst present.

The initial state I is an n component vector with the i'th component I_i being the mass fraction of catalyst in compartment i. Note $\sum_{i=1}^{n} I_i = 1$ since the tank is a closed system as far as the catalyst is concerned. The distribution of mass fractions of catalyst over the compartments after a sequence of controls $X = i_1 \ldots i_m$ is just

$$I \cdot A(i_1) \cdot \ldots \cdot A(i_m) \quad = \quad I \cdot A(X) \quad .$$

That is, $(I \cdot A(X))_i$ is the mass fraction of catalyst in compartment i after starting with initial distribution I of catalyst fractions over compartments and the string of control inputs $X = i_1 \ldots i_m$.

The total end product from the tank is the sum of the outputs from each compartment: $\sum_{i=1}^{n} (I \cdot A(X))_i F_i$ which can be written $I \cdot A(X) \cdot F$ in matrix notation. This expression has the same form as the expectation of output for the probabilistic slot-machine, but there are no overt probabilities involved here. The mass fractions of catalyst play the same role as the probabilities in the first example. However, the output will still be written like an expectation as $E_A(X)$.

The total end product accumulated, T_X, for the string of controls X from time t_0 to time $t_0 + m$ is given by adding the output from each substring, i.e.,

$$T_X \quad = \quad E_A(i_1) \; + \; E_A(i_1 i_2) + \ldots + E_A(i_1 i_2 \ldots i_m) \quad .$$

1. DETERMINING WHETHER A PROBABILISTIC SEQUENTIAL MACHINE IS EXPECTATION EQUIVALENT TO A FINITE DETERMINISTIC MACHINE

1.1 THE CONCEPT OF EXPECTATION EQUIVALENCE

In the two models discussed in the introduction, the expected value of output, $E_A(x)$, played an important role in the physical interpretations. Let us repeat the definition of the expected value of output.

Definition 1.1: The expected value of output for a probabilistic sequential machine A is given by

$$E_A(x) \quad = \quad I \cdot A(x) \cdot F \quad \text{for } x \text{ in } \Sigma^* \quad .$$

Definition 1.2: Machines A and A' are expectation equivalent, written $A \equiv {}_E A'$, if

$$E_A(x) \quad = \quad E_{A'}(x) \quad \text{for all } x \text{ in } \Sigma^*$$

Recall from example 0.2 that $E_A(x)$ was the actual output of the chemical cell and not an expectation. Hence the basic concept of expectation equivalence is analogous to the definition of behavioral equivalence of Burks for the model 0.2. However for example 0.1, the slot-machine, expectation equivalence is not the generalization of this kind of behavioral equivalence. The concept of indistinguishability discussed in Chapter 3 seems to be the appropriate generalization of this kind of behavioral equivalence. Let us now turn to an example to show how proper coding of the outputs could make the concept of expected value of output relevant to an unreliable digital computer.

Example 1.1. Proper choice of output code can make the expected value of output relevant to the study of real computers. We encode the output so $E_A(x)$ is approximately the code for output for string x. Then expectation equivalent machines have nearly the same input-output behavior when one averages it over a large number of programs.

Suppose from some machine A we have

$$\begin{array}{lcllllll} IA(x) & = & (.0000, & .0625, & .8750, & .0625, & .0000, & \ldots) \\ IA(y) & = & (.8750, & .0625, & .0000, & .0000, & .0625, & \ldots) \\ IA(z) & = & (.0000, & .0000, & .0000, & .1250, & .8750, & \ldots) \end{array}$$

and

$$F^T = (\ T, \quad \rlap{T}A, \quad A, \quad A*, \quad *, \quad \ldots)$$

with the intent that

x causes an "A" as output
y causes a "T" as output
z causes a "*" as output

We can recode the output symbols by the following ($F^{T'}$ is F^T recoded)

$$F^{T'} = (100,\ 011,\ 010,\ 001,\ 000,\ \ldots)$$

and

$$\begin{array}{lcll} E_A(x) & = & 010_2 & \text{which is the code for A} \\ E_A(y) & = & 100_2 & \text{which is the code for T} \\ E_A(z) & = & (.001)_2 & \text{which is not a code, but} \end{array}$$

if decoding is used which picks the closest code number, z is associated with output "*".

A more careful choice of code numbers could have made each expectation equal to a code, simplifying the decoding problem.* However in a practical situation, only a sample expectation to be decoded can be obtained and a more elaborate statistical decision rule than just comparing for equality must be used in decoding the output symbol.

*Proof: Let X_i be the code weight. $(X_i = F_i)$ and $I \cdot A(z_i) = (P_{i_1}, P_{i_2}, \ldots, P_{i_n})$

$i = 1,2,\ldots,n$

The condition that $E_A(z_i) = X_i$ $i = 1,2,\ldots,n$ implies that

$$\begin{array}{ccc} P_{11}X_1 + P_{12}X_2 + \ldots + P_{1n}X_n & = & X_1 \\ \vdots & & \vdots \\ P_{n1}X_1 + P_{n2}X_2 + \ldots + P_{nn}X_n & = & X_n \end{array}$$

or equivalently

$$P'X = \begin{pmatrix} P_{11}-1 & P_{12} & \ldots & P_{1n} \\ P_{21} & P_{22}-1 & \ldots & P_{2n} \\ \vdots & & & \vdots \\ P_{n1} & \ldots & \ldots & P_{nn}-1 \end{pmatrix} \begin{pmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_n \end{pmatrix} = 0$$

which has a non-zero solution iff Determ.(P') = 0.

By definition an eigenvalue of a matrix M is some number λ_i such that

$$\text{Determ.}\begin{pmatrix} m_{11}-\lambda_i & m_{12} & \ldots & m_{1n} \\ m_{21} & m_{22}-\lambda_i & \ldots & m_{2n} \\ m_{n1} & \ldots & \ldots & m_{nn}-\lambda_i \end{pmatrix} = 0$$

For any stochastic matrix, 1 is an eigenvalue.

Hence Determ.(P') = 0 is always true for any choice of probabilities and the result follows.

In order for the encoding to be unique we also need $X_i \neq X_j$, $i \neq j$ but conditions on the probabilities for this to occur will not be considered here.

Example 1.2.

Machines A and A' which are expectation equivalent: $A \equiv_E A'$

$$IA(x)F = I'A'(x)F' \quad \forall x \in \Sigma^*$$

$$A = < I, A(0), A(1), F > \text{ and } A' = < I, A'(0), A'(1), F >$$

$$A(0) = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/4 & 1/4 \\ 1/4 & 0 & 3/4 \end{pmatrix} \qquad A(1) = \begin{pmatrix} 3/5 & 1/5 & 1/5 \\ 1/5 & 4/5 & 0 \\ 4/5 & 1/5 & 0 \end{pmatrix}$$

$$A'(0) = \begin{pmatrix} 1 & 0 & 0 \\ 5/8 & 0 & 3/8 \\ 0 & 1/2 & 1/2 \end{pmatrix} \qquad A'(1) = \begin{pmatrix} 7/10 & 0 & 3/10 \\ 3/5 & 0 & 2/5 \\ 9/10 & 0 & 1/10 \end{pmatrix}$$

$$F = F' = \begin{pmatrix} 7 \\ 5 \\ 3 \end{pmatrix}.$$

These machines are expectation equivalent from any initial probability distribution, I, over states.

The previous example shows that two machines can have very different switching matrices and still be expectation equivalent. Frequently, studies of Markov processes are concerned with the location of the zeros in the transition matrices. The example shows that the locations of zeros in the transition matrices is not the only relevant factor in the study of expectation equivalence. Since the graph theoretic properties of the transition matrices, such as the accessibility of a state, depend on where the zeros are, one would not expect a purely graph theoretic approach to be very fruitful in the study of this problem. Hence some of the tools of linear algebra will be used in addition to the above approaches.

1.2 THE REDUCTION RELATION R_F

In this section a congruence relation, R_F, will be defined so that a quotient machine can be constructed. States of the quotient machine will correspond to the distinct values of expectation which occur for input strings. If the rank of R_F happens to be finite, the machine constructed has a finite number of states. By attaching a deterministic output device to each state of the constructed machine, the expectation equivalent deterministic machine is obtained.

If the rank of R_F is finite, some class of the relation must contain infinitely many strings. A necessary condition for R_F to be finite in rank is that it be non-trivial, i.e. at least two different strings are contained in some class. This necessary condition produces strong constraints on the form of the symbol matrices of such probabilistic machines.

Definition 1.3: The reduction relation R_F is given by

$$x\ R_F\ y \text{ iff } E_A(x) = E_A(y)\ \&\ E_A(xz) = E_A(yz)\ \forall z \in \Sigma^*\ \forall I \in S$$

If Σ contains Λ, a semigroup identity, the definition reduces to

$$x\ R_F\ y \text{ iff } E_A(xz) = E_A(yz)\ z \in \Sigma^*,\ \forall I \in S\ .$$

R_F is a congruence relation on Σ^* because of the reflexivity, transitivity and symmetry of "=" and the substitution property in its definition.

In order to discuss congruence relations between stochastic matrices which may not be generated by strings of symbol matrices a matrix congruence analogous to R_F will be defined.

Definition 1.4: The matrix reduction relation R_M between n x n stochastic matrices B and B':

$$B\ R_M\ B' \text{ iff } IBF = IB'F' \text{ and there exist machines}$$

A and A' such that $IBA(z)F = IB'A'(z)F'$ for all $I \in S$, for all $z \in \Sigma^*$.

Hence two strings x and y which are in the same class of the relation R_F will have equal expectations from any initial state of the machine and will continue to have equal expectations for any finite input continuation z. As far as expectation of output is concerned, the behavior of the machine A is the same after either string x or string y. Returning to Example 1.1, we can interpret x and y as program segments which produce the same final output code and from which any continuation will give the same output code. If intermediate outputs are suppressed, x and y in R_F can be regarded as equivalent microprograms in the machine A.

1.3 CONSTRUCTION OF THE QUOTIENT MACHINE

Definition 1.5: The equivalence class of x' of R, a congruence relation, is given by

$$R[x'] = \{x : x \; R \; x'\}$$

It is a well known result [10] of automata theory that given a right congruence relation R on Σ^*, one can construct a quotient automaton with no output T(R)

$$T(R) = < a, S, M >$$

where

$$a = R[\Lambda]$$

$$S = \{R[x] : x \in \Sigma^*\}$$

M is a function from S x Σ into S such that

$$M(R[x],\sigma) = R[x\sigma] \quad x \in \Sigma^*; \; \sigma \in \Sigma$$

Definition 1.6: Let $\beta \subseteq \Sigma^*$. A congruence R refines β if

$$x \; R \; y \Rightarrow x \in \beta \text{ iff } y \in \beta \; .$$

Theorem 1.1 (Nerode)

Let β be a subset of Σ^*. β is the behavior of a finite (deterministic) automaton $A = < T(R), \mathcal{F} >$ over Σ where $\mathcal{F} = \{R[x] : x \in \beta\}$ iff there exists a right congruence relation R of finite rank which refines β.

Theorem 1.2

If the congruence relation R_F has finite rank, then for any λ there is a finite deterministic automaton A' such that the tapes accepted by A' are $T(A,\lambda)$.

Proof: Let $\beta = T(A,\lambda) = \{x : E_A(x) \geq \lambda\}$. Note that R_F refines β i.e. $x R_F y \Rightarrow x \in T(A,\lambda)$ iff $y \in T(A,\lambda)$. If R_F has finite rank, by definition $\{R_F[x]\}$ has a finite number of members. Using theorem 1.1 we construct

$$T(R_F) = < a, S, M > \quad \text{and}$$

$$A' = < a, S, M, \mathcal{F} > \text{ which accepts } T(A,\lambda)$$

Q.E.D.

Definition 1.7: $rp_A(x)$ is the response of A to input string x. If A is deterministic, $rp_A(x)$ is the state of A after an input of x. If A is probabilistic, $rp_A(x)$ is a random variable taking on values which are states.

We use the above construction to give a sufficient condition for the reduction of a probabilistic sequential machine into an expectation equivalent finite deterministic machine whose output function is either a constant C(s) for each state s or a random device $O_A^R(s)$ with expectation $E(O_A^R(s)) = C(s)$.

Theorem 1.3

The reduction relation R_F defined by a probabilistic machine A has finite rank if and only if there exists a finite deterministic machine A' with a deterministic output $O_{A'}$ such that $O_{A'}(rp_{A'}(x)) = E_A(x)\ \forall x \in \Sigma^*$

Proof: (sufficiency) By theorem 1.1 let A' = < a, S, M, ϕ > where ϕ is the empty set. Note any congruence R refines ϕ vacuously. We attach an output function $O_{A'}$ to elements of S.

$$O_{A'}(s) = E_A(x) \quad s = R_F[x] \quad .$$

For a deterministic machine, M is extended to M^* which operates on strings rather than symbols by

$$M^*(s,\sigma) = M(s,\sigma) \qquad s \in S \quad \sigma \in \Sigma$$

$$M^*(s,\sigma x) = M^*(M^*(s,\sigma),x) \qquad x \in \Sigma^*$$

We note that $M^*(a,x) = rp_{A'}(x)$ so we need show only that $rp_{A'}(x) = R_F[x]$. Let $x = i_1 i_2 \ldots i_m$ for $i_j \in \Sigma$; $j = 1,2,\ldots,m$.

$$\begin{aligned} rp_{A'}(x) = M^*(z,x) &= M^*(M^*(a,i_1),i_2\ldots i_m) \\ &= M^*(M\ (a,i_1),i_2\ldots i_m) \\ &= M^*(M(R_F[\Lambda],i_1),i_2\ldots i_m) \\ &= M^*(R_F[\Lambda i_1],i_2\ldots i_m) \\ &= M^*(M(R_F[i_1],i_2),i_3\ldots i_m) \\ &\quad \ddot{} \qquad\qquad \ddot{} \\ &= R_F[i_1 i_2\ldots i_m] = R_F[x] \end{aligned}$$

Hence the constructed sequential machine is A' = < a, S, M, $O_{A'}$ >

(necessity)

Given A' = < a, S, M, $O_{A'}$ > such that

$$O_{A'}(rp_{A'}(x)) = E_A(x) \quad \forall x \in \Sigma^*$$

$$O_{A'}(rp_{A'}(xz)) = E_A(xz) \quad \forall z \in \Sigma^*$$

Let $rp_{A'}(x) = S_x \quad x \in \Sigma^*$

Define

$$S_x\ R_0\ S_y \ \text{ iff } \ x\ R_F\ y$$

Let n' be the cardinality of S — finite.

$$\text{rank } R_O = \text{rank } R_F$$

$$\text{rank } R_O \leq n'$$

Hence rank R_F is finite. Q.E.D.

Instead of the deterministic function $O_{A'}$, a random device $O^R_{A'}(s)$ such that $E(O^R_{A'}(s)) = E_A(x)$ could have been used in the construction .

1.4 THE PARTITION OF THE SET OF ACCESSIBLE STATE DISTRIBUTIONS INDUCED BY R_F

Definition 1.8: $V(A) = \{IA(x) : x \in \Sigma^*\}$ — the set of all stochastic vectors which can occur as distributions over the states of A. We sometimes call V(A) the "state vectors accessible in A".

Definition 1.9: A set of vectors $V = \{v_1,v_2,\ldots\}$ is convex if for any finite set of indices I $c_i \geq 0$ and $\sum_{i\in I} c_i = 1 \rightarrow \sum_{i\in I} c_i v_i \in V$. The convex closure of a set of vectors V, written $V^+ = \{v' : v' = \sum_{i\in I} c_i v_i$ and $\sum_{i\in I} c_i = 1$ and $c_i \geq 0$ and $v_i \in V\}$.

Theorem 1.4

If R_F has finite rank r, there exists a partition $II = (II_1,\ldots,II_r)$ on V(A) and an integer valued function g(1,m) such that

$$II_i A(\sigma) = II_{g(i,\sigma)} \quad i = 1,\ldots,r; \quad \sigma \in \Sigma$$

Proof: We use R_F to induce an equivalence on the set of stochastic vectors accessible by the machine.

Since R_F is of finite rank, we form a set of an arbitrary distinct representative from each congruence class, say $x_1,\ldots,x_r$ where $x_i \neq x_j$

$i = 1,2,\ldots,r;\ j < i.$

Define $II_j = \bigcup_{x \in R_F[x_j]} \{IA(x)\}$

We show that $(II_1,\ldots,II_r)$ is a partition of $V(A)$

Let $W = \bigcup_{i=1}^{r} II_i$

$$IA(x') \in W \Rightarrow IA(x') \in V(A)$$

$$\begin{aligned} IA(x') \in V(A) &\Rightarrow x' \in R_F[x_k] \text{ for some } k = 1,\ldots,r \\ &\Rightarrow IA(x') \in II_k \text{ for some } k = 1,\ldots,r \\ &\Rightarrow IA(x') \in W \end{aligned}$$

Hence

$$W = \bigcup_{i=1}^{n} II_i = V(A)$$

We show

$$II_i \cap II_j = \emptyset \quad i \neq j$$

suppose that

$$II_i \cap II_j = \{v_y,\ldots .\}$$

where

$$v_y = IA(y)$$

$$IA(y) \in II_i \Rightarrow y \in R_F[x_i] \Rightarrow y\ R_F\ x_i$$

$$IA(y) \in II_j \Rightarrow y \in R_F[x_j] \Rightarrow y\ R_F\ x_j$$

Hence we get

$$y\ R_F\ x_i \Rightarrow x_i\ R_F\ y \text{ by symmetry}$$

and transitivity of R_F gives

$$x_i\ R_F\ x_j \Rightarrow x_i \in R_F[x_j]$$

But since x_i and x_j are representatives and there is only one representative from each class

$$x_i = x_j \quad i \neq j$$

which is a contradiction.

Finally we show there exists an integer valued function $g(i,\sigma)$ such that

$$\Pi_i A(\sigma) = \Pi_{g(i,\sigma)} \quad \sigma \in \Sigma$$

$$v_1 \in \Pi_i \Rightarrow v_1 = IA(w_1) \quad \text{for some } w_1 \in \Sigma^*$$

$$v_1 A(\sigma) = IA(w_1)A(\sigma) = IA(w_1\sigma) \in \Pi_j$$

for some j as has been shown above

$$v_2 \in \Pi_i \Rightarrow v_2 = IA(w_2) \quad \text{for some } w_2 \in \Sigma^*$$

$$v_2 A(\sigma) = IA(w_2\sigma) \in \Pi_j$$

since elements of R_F have the substitution property, i.e.

$$w_1 \; R_F \; x_i \Rightarrow w_1\sigma \; R_F \; x_i\sigma \quad \sigma \in \Sigma$$

$$w_2 \; R_F \; x_i \Rightarrow w_2\sigma \; R_F \; x_i\sigma \quad \sigma \in \Sigma$$

$x_i\sigma$ is an element of a class with representative x_j for some j and depends only on x_i and σ. So there is a function $g(1,m)$ such that

$$g(i,\sigma) = j \quad \sigma \in \Sigma \qquad \text{Q.E.D.}$$

1.5 NECESSARY AND SUFFICIENT CONDITIONS THAT STRINGS BE IN THE SAME R_F CLASS

The relation R_F has occupied an important place in the development of this theory. We now study the structure of the matrices of strings which are in the same R_F class.

Definition 1.10: A relation R is non-trivial if there exist x and y in the domain of R with $x \neq y$ such that $x \; R \; y$.

Definition 1.11: The **kernel of F** = Kern. (F)

$$= \{v \in R^n : v \cdot F = 0\}$$

where R is the set of reals.

Definition 1.12: The **span** of a set of vectors $\{v_1,\ldots,v_r\}$ is denoted by

$$< \{v_1,\ldots,v_r\} > = \left\{ \sum_{i=1}^{r} c_i v_i \ \forall c_i \in R \right\}$$

Theorem 1.5

A necessary and sufficient condition for x and y to be in the same class of the reduction relation R_F is:

$$(x,y) R_F \Leftrightarrow \exists \text{ a subspace } U \text{ of Kern. } (F) \text{ such that}$$

$$\text{(i) } UA(z) \subset \text{Kern. } (F) \quad z \in \Sigma^*$$

$$\text{(ii) } A(x) = A(y) + \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \text{with } u_i \in U \quad i = 1,\ldots,n$$

Proof:

$$(x,y) \in R_F \Rightarrow IA(xz)F = IA(yz)F \quad \forall I \in S \quad \forall z \in \Sigma^*$$

hence

$$A(x)F = A(y)F$$

since

$$S = \{(1,0,\ldots,0),\ldots,(0,\ldots,0,1)\} \text{ and } \Lambda \in \Sigma^*$$

by linear algebra

$$A(x) = A(y) + \begin{vmatrix} h_1 \\ \vdots \\ h_n \end{vmatrix} \text{ where } h_i \in \text{Kern. } (F), \quad i = 1,2,\ldots,n$$

multiplying by A(z)

$$A(x)A(z) = A(y)A(z) + \left|\begin{matrix} h_1 \\ \vdots \\ h_n \end{matrix}\right| A(z) \quad \forall x \in \Sigma^*$$

$$A(xz) = A(yz) + \left|\begin{matrix} h_1 \\ \vdots \\ h_n \end{matrix}\right| A(z)$$

we can multiply by any initial state distribution I so

$$IA(xz)F = IA(yz)F + I \left|\begin{matrix} h_1 \\ \vdots \\ h_n \end{matrix}\right| A(z)F \quad I \in S^+$$

But since x and y are in R_F

$$IA(xz)F = IA(yz)F \quad \forall z \in \Sigma^* \quad I \in S^+$$

Hence

$$I \left|\begin{matrix} h_1 \\ \vdots \\ h_n \end{matrix}\right| A(z)F = 0 \Rightarrow h_i\ A(z) \in \text{Kern.}(F), \quad i = 1,2,\dots,n.$$

Let $U = < h_1,\dots,h_n >$

We get $UA(z) \subset \text{Kern.}(F) \quad \forall z \in \Sigma^*$

Notation: we denote by A_i the i'th row of the matrix A.

Let $H = \left|\begin{matrix} h_1 \\ \vdots \\ h_n \end{matrix}\right|$ where $h_i \in U \subset \text{Kern.}(F), \quad i = 1,2,\dots,n.$

$$* \quad A(x) = A(y) + H$$

multiplying the equality by I on the left and F on the right we obtain

$$IA(x)F = IA(y)F + I \cdot \begin{pmatrix} h_1 \cdot F \\ \vdots \\ h_n \cdot F \end{pmatrix} \quad I \in S^+$$

but $h_iF = 0$ since $h_i \in \text{Kern.}(F) \quad i = 1,\dots,n$

$$IA(x)F = IA(y)F$$

using * again and the same argument we get

$$\begin{aligned} A(xz) &= A(yz) + HA(z) \\ IA(xz)F &= IA(yz)F + IHA(z)F \\ &= IA(yz)F \end{aligned}$$

since $HA(z) = \begin{vmatrix} u_i \\ \vdots \\ u_n \end{vmatrix}$ where $u_i \in$ Kern.(F) Q.E.D.

We now simplify the restriction (i) of Theorem 1.6, to symbol matrices rather than string matrices.

Theorem 1.6

Let $U = < \bigcup_{x \in \Sigma^*} \{A(x)_i - A(y)_i\} : i = 1,\ldots,n$ for x,y such that $(x,y) \in R_F >$ then

$$U \cdot A(z) \subset \text{Kern.}(F) \Leftrightarrow [\exists V \text{ a subspace of } R^n:$$

$$\text{(i) } UA(i) \subset V : \forall i \in \Sigma$$

$$\text{(ii) } VA(i) = V \subset \text{Kern.}(F) \quad \forall i \in \Sigma]$$

Proof:

$$UA(z) \subset \text{Kern.}(F)$$

$$\text{Let } V = < \{u \cdot A(z); \quad u \in U, \quad z \in \Sigma^*\} >$$

$$\begin{aligned} VA(i) &= \{uA(z)A(i); \quad u \in U, \quad z \in \Sigma^*\} \\ &= V \end{aligned}$$

Consider an arbitrary $v \in V$. There must be some set of indexes J and constants c_j such that:

$$v = \sum_{j \in J} c_j u_j A(z_j) \text{ by definition of V.}$$

$$v \cdot F = \left(\sum_{j \in J} c_j u_j A(z_j) \right) \cdot F$$

$$= \sum_{j \in J} c_j (u_j A(z_j) F)$$

But

$$u_j A(z_j) F = 0 \quad \text{by} \quad UA(z) \subset \text{Kern.}(F)$$

so

$$v \cdot F = 0$$

Hence

$$V \subset \text{Kern.}(F)$$

$$UA(z) \quad V \text{ already shown}$$

$$\therefore UA(z) \subset \text{Kern.}(F)$$

Q.E.D.

Definition 1.13: A subspace V is invariant under a set of linear transformations

$$\{T_i : i = 1,2,\ldots,m\} \quad \text{if} \quad V \cdot T_i = V \quad i = 1,2,\ldots,m \ .$$

Using Theorems 1.5 and 1.6, we get the following directly:

Theorem 1.7

Strings x and y are in the same class of R_F if and only if there exists a subspace V of Kern.(F) such that

(i) V is invariant under $\{A(i) ; \forall i \in \Sigma\}$

(ii) $A(x) = A(y) + H$ where $H_i \in V \quad i = 1,\ldots,n$

1.6 NECESSARY AND SUFFICIENT CONDITIONS THAT R_F BE NON-TRIVIAL

A very weak necessary condition that R_F have finite rank is that it at least be non-trivial. From theorem 1.7 it is immediate that:

Theorem 1.8

The reduction relation R_F is non-trivial $\Longleftrightarrow \exists$ a subspace V of Kern.(F) such that

$$\text{(i)}\quad V \text{ is invariant under } \{A(i);\ \forall i \in \Sigma\}$$

$$\text{(ii)}\quad A(x) = A(y) + H \text{ where } H_i \in V \quad i = 1,\dots,n$$

$$\text{(iii)}\quad x \neq y \quad .$$

Hence we now know that given strings x and y in R_F, the difference between the rows of the matrices A(x) and A(y) must be elements of a subspace V which has special properties. Namely V must be invariant under all symbol matrices and contained in the kernel of the output vector.

Theorem 1.9

A necessary and sufficient condition that R_F be non-trivial is that $A(i) : \forall i \in \Sigma$ be reducible for the same change of basis to V. In other words, there exists a linear transformation W of the state vectors S to a basis for V such that

$$W^{-1}A(i)W = \begin{array}{c} \text{basis for } V \quad \\ \downarrow \qquad\quad \\ \left[\begin{array}{cc} A_1^i & 0 \\ A_2^i & A_3^i \end{array}\right] \end{array}$$

Where 0 denotes a submatrix of zeros and A_1^i, A_2^i, and A_3^i are submatrices which for all i in Σ have the same number of columns and rows.

Proof: By theorem 1.7 and standard matrix theory [see Jacobson, Lectures in Abstract Algebra, V. II: Linear Algebra, Van Nostrand, New York, 1952, pp. 116-117].

Consequently, Theorem 1.9 gives us a matrix reformulation of the statement that R_F be non-trivial.

Example 1.3. We now show a probabilistic sequential machine A which illustrates theorems 1.3 and 1.7. The method by which this example was generated will be discussed in a latter report.

$$A = < I, A(0), A(1), F >$$

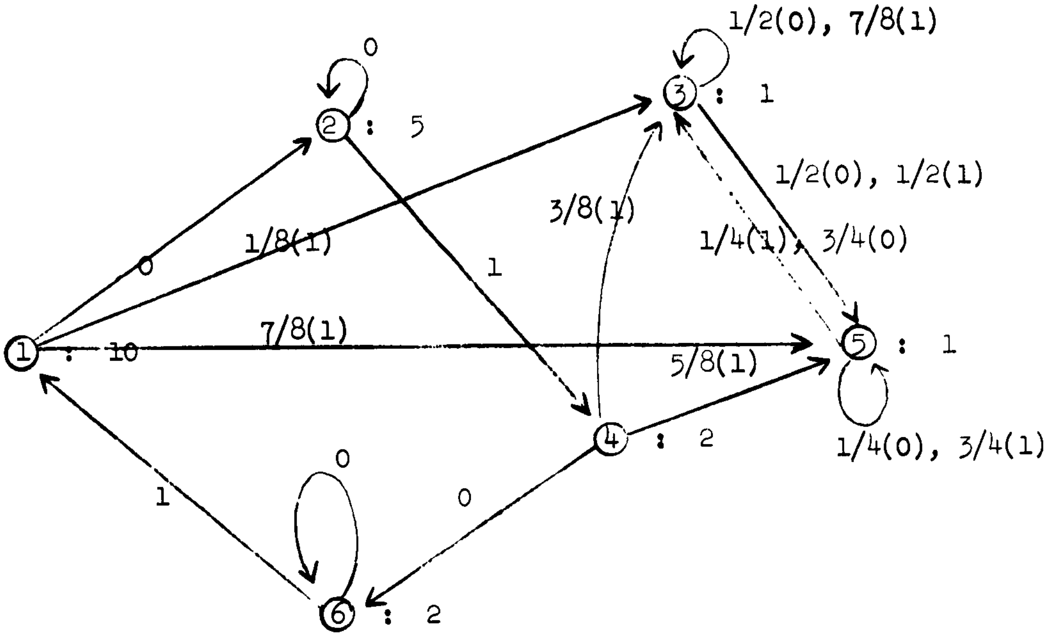
where

$$I = (8/10, 1/10, 1/10, 0, 0, 0)$$

$$A(0) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad F = \begin{pmatrix} 10 \\ 5 \\ 1 \\ 2 \\ 1 \\ 2 \end{pmatrix}$$

$$A(1) = \begin{pmatrix} 0 & 0 & 1/8 & 0 & 7/8 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 4/8 & 0 & 4/8 & 0 \\ 0 & 0 & 3/8 & 0 & 5/8 & 0 \\ 0 & 0 & 2/8 & 0 & 6/8 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The state diagram for A:



where the following labeling conventions are used.

p(K) : pϵ[0,1]; K ϵ Σ means probability of transition of p via symbol K.

ℓ : F_ℓ : Output of F_ℓ occurs when the machine is in state ℓ.

$$0 \underset{P_2(K_2)}{\overset{P_1(K_1)}{\Longrightarrow}} 0 : \text{is replaced by } 0 \xrightarrow{P_1(K_1),\ P_2(K_2)} 0$$

We note that

$$00R_F0$$

since:

$$A(00) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3/4 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3/4 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5/8 & 0 & 3/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 9/16 & 0 & 7/16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

which gives

$$(A(00) - A(0))F = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1/8 & 0 & -1/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3/16 & 0 & 3/16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 10 \\ 5 \\ 1 \\ 2 \\ 1 \\ 2 \end{pmatrix} = (0,0,0,0,0,0)$$

Hence $A(00)F = A(0)F$ or $IA(00)F = IA(0)F$ for all I.

Furthermore, for all $P \in [0,1]$

$$(0,\ 0,\ P,\ 0,\ 1-P,\ 0)A(0) = (0,\ 0,\ P,\ 0,\ 1-P,\ 0)$$

$$(0,\ 0,\ P,\ 0,\ 1-P,\ 0)A(1) = (0,\ 0,\ P,\ 0,\ 1-P,\ 0)$$

that is

$$W = < \{(0,\ 0,\ P,\ 0,\ 1-P,\ 0)\} >$$

is invariant under the symbol matrices $A(0)$ and $A(1)$.

$$V = < \{(0,\ 0,\ P,\ 0,\ -P,\ 0) > \subset W \quad \text{and} \quad \begin{array}{l} VA(0) = V \\ VA(1) = V \end{array}$$

Hence for $z \in \Sigma^*$

$$(A(00) - A(0))A(Z) = C_z \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1/8 & 0 & -1/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3/16 & 0 & 3/16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = D$$

where C_z is a constant depending on the string z

and

$$(A(00) - A(0))A(z)F = DF = 0$$

consequently $\forall z \in \Sigma^* \quad \forall I \in S^+$

$$IA(00)A(z)F = IA(0)A(z)F$$

or

$$E_A(00z) = E_A(0z), \text{ which shows } 00R_F0.$$

By the same method one can show that

$$10\ R_F\ 1 \quad 011\ R_F\ 11 \quad 01011\ R_F\ 11 \quad 111\ R_F\ 11 \quad 01010\ R_F\ 0$$

and all strings are in the classes

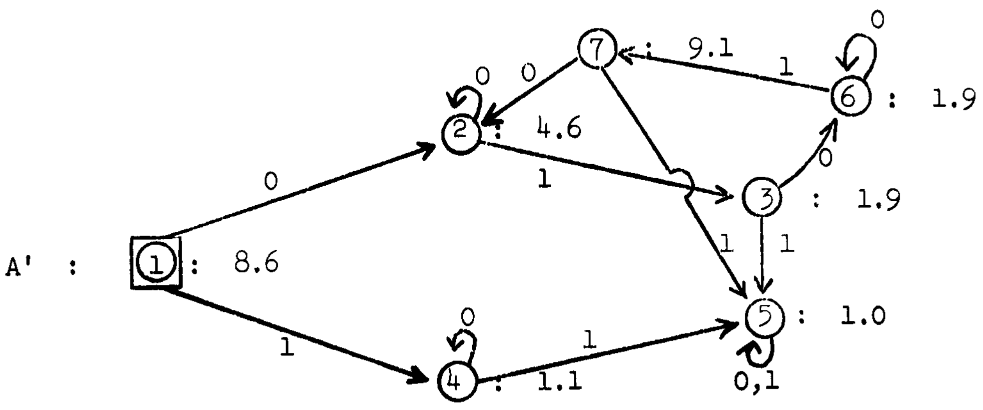
$$R_F(\Lambda), \quad R_F[0], \quad R_F[1], \quad R_F[11], \quad R_F[01], \quad R_F[010], \quad R_F[0101]$$

which means that R_F has finite rank.

We compute the expectations and construct the expectation equivalent deterministic machine A'. Note that the values of expectation depend on the initial state I.

$$\begin{aligned} E_A(\Lambda) &= IA(\Lambda)F = IF = 8.6 \\ E_A(0) &= (0,\ 9/10,\ 1/20,\ 0,\ 1/20,\ 0)F = 4.6 \\ E_A(1) &= (0,\ 0,\ 3/20,\ 2/20,\ 15/20,\ 0)F = 1.1 \\ E_A(01) &= (0,\ 0,\ 3/80,\ 72/80,\ 5/80,\ 0)F = 1.9 \\ E_A(10) &= (0,\ 0,\ 3/20,\ 0,\ 15/20,\ 2/20)F = 1.1 = E_A(1) \text{ (since } 10R_F1) \\ E_A(11) &= (0,\ 0,\ 9/40,\ 0,\ 31/40,\ 0)F = 1.0 \\ E_A(010) &= 1.9 \\ E_A(0101) &= 9.1 \end{aligned}$$

Hence the expectation equilvalent deterministic machine is



We note that A' has 7 states while A has just 6 states. The deterministic cycle 0101 appears in both machines

2. DETERMINING WHETHER A PROBABILISTIC SEQUENTIAL MACHINE IS N-MOMENT EQUIVALENT TO A MACHINE WITH DETERMINISTIC SWITCHING AND RANDOM OUTPUTS

In this section the concept of expectation equivalence is generalized to N-moment equivalence. A congruence relation R_F^N is defined which partitions the input strings into classes whose members all produce the same expectation and first N-1 central moments in a given machine. If R_F^N has finite rank, a finite quotient machine can be constructed which is deterministic with each state corresponding to a congruence class. Each state can be connected to a random device having the same expectation and N-1 moments as the class represented by the state, giving a deterministic machine with random outputs. The deterministic machine constructed is N-moment equivalent to the probabilistic machine. After the first theorem concerning a necessary and sufficient condition that two strings be in the same R_F^N class, it is obvious that a simple substitution gives generalizations of the results of section 1 and they are presented in this section without proofs.

2.1 DISTRIBUTION EQUIVALENCE: $\equiv_D$

The random variable structure of probabilistic sequential machines will be investigated in this section.

Definition 2.1: $O_A^*(x)$ = the output random variable of the machine A after a string x has occurred as input.

The distribution of $O_A^*(x)$ is IA(x) and values of $O_A^*(x)$ are the entries of F.

Definition 2.2: A and A' are distribution equivalent written $A \equiv_D A'$ if for $J_A = \{j : (IA(x)_j F_j \neq 0\}$ there is a 1-1 map h between J_A and $J_{A'}$ such that

$$IA(x)_{h(j)} = I'A'(x)_j \quad j \in J_{A'} \quad x \in \Sigma^*$$

$$F_{h(j)} = F'_j \quad j \in J_{A'}$$

Distribution equivalence corresponds to the conventional definition of equivalence for discrete random variables except for random variables $F_i \neq F_j$ for $i \neq j$.

Referring back to example 0.2 note that two chemical cells are distribution equivalent if (1) We neglect those partitioned areas which have either zero efficiency or a zero fraction of the catalyst. (2) Of the remaining partitioned areas there is a correspondence between the partitioned areas of one cell and the other such that corresponding areas have the same fraction of catalyst regardless of the sequence of controls entering the cells. (3) Corresponding partitioned areas have the same efficiencies.

2.2 MOMENTS OF THE OUTPUT RANDOM VARIABLE

Definition 2.3: Let

$$F = \begin{vmatrix} F_1 \\ \vdots \\ F_n \end{vmatrix} \quad F_i \in R \quad i = 1,2,\dots,n$$

call

$$(F^k) = \begin{vmatrix} F_1^k \\ \vdots \\ F_n^k \end{vmatrix} .$$

Then the i'th central moment of $O_A^*(x)$ is

$$\mu_i^A(x) = E_A[(O_A^*(x)-E_A(x))^i] \quad i = 2,3,\ldots$$

Sometimes $\mu_1^A(x) = E_A(x)$ will be used informally.

Theorem 2.1

$$\mu_i^A(x) = \sum_{k=0}^{i} \binom{i}{k}(-1)^k IA(x)(F^{i-k})E_A(x)^k \quad i = 2,3,\ldots$$

Proof: By the binominal theorem

$$\mu_i^A(x) = E_A\left[\sum_{k=0}^{i} (-1)^k O_A^*(x)^{i-k}\, E_A(x)^k \binom{i}{k}\right]$$

To compute the expectation of the discrete random variable $O_A^*(x)^{i-k}$ note it has the same distribution as $O_A^*(x)$ but takes on values $F_1^{i-k},\ldots,F_n^{i-k}$ for $i \neq k$

$$\mu_i^A(x) = \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} E_A[O_A^*(x)^{i-k}]E_A(x)^k + (-1)^i E_A(x)^i$$

$$= \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} \cdot IA(x)(F^{i-k})E_A(x)^k + (-1)^i E_A(x)^i \quad \text{Q.E.D.}$$

2.3 SPECIAL PROPERTIES OF RABIN PROBABILISTIC AUTOMATA

Definition 2.4: A Rabin probabilistic automaton [4] is a probabilistic sequential machine such that $F_i = 0$ or $F_i = 1 \quad i = 1,2,\ldots n$.

We now observe that Rabin probabilistic automata have rather special features as far as the random variable of the output is concerned.

Corrollary 2.1: For a Rabin probabilistic automaton A

$$\mu_i^A(x) = \sum_{k=0}^{i-1} (-1)^k \binom{i}{k} E_A(x)^{k+1} + (-1)^i E_A(x)^i \quad i = 2,3,\ldots$$

Proof: $F_i = 0$ or 1 hence

$$(F^{i-k}) = F \quad \text{for} \quad i \neq k$$

and the result from Theorem 2.1.

Corrollary 2.2: If $E_A(x) = E_A(y)$ for some Rabin probabilistic automaton A, then all central moments for x and y are equal also, i.e.

$$\mu_i^A(x) = \mu_i^A(y) \quad \text{for} \quad i = 2,3,\ldots$$

Note: for i = 2 we get the variances of the outputs are equal.

Corrollary 2.3: If two Rabin probabilistic automaton A and A' are expectation equivalent then

$$\mu_i^A(x) = \mu_i^{A'}(x) \quad i = 2,3,\ldots \quad \forall x \in \Sigma^* .$$

2.4 THE CONCEPT OF N-MOMENT EQUIVALENCE: $\equiv_N$

Even if two machines are expectation equivalent, the statistics of their behavior may be so different that for many purposes we would not want to consider the machines behaviorally equivalent. Returning to example 0.1, two slot-machines can be expectation equivalent, meaning that the average payoff is the same for both, but one can be much more desirable than the other for a player of limited resources. For a player with limited resources might have a far longer average time until "gambler's ruin" on one machine rather than the other. Hence in order to lump those machines in the same class whose statistics of behavior are somewhat alike, we introduce the notion of N-moment equivalence.

Definition 2.5: Probabilistic sequential machines A and A' are N-moment equivalent, written $\equiv_N$, if

$$E_A(x) = E_{A'}(x)$$
$$\mu_i^A(x) = \mu_i^{A'}(x) \quad i = 2,\dots,N \quad \text{for all } x \text{ in } \Sigma^*$$

2.5 THE RELATIONSHIP BETWEEN $\equiv_D$ AND $\equiv_N$

Theorem 2.2

For probabilistic sequential machines A and A'

$$A \equiv_D A' \Rightarrow A \equiv_N A' \text{ for all finite N}$$

Proof: Distribution equivalence means there exists an h such that

$$F_{h(i)} = F_i'$$

$$(IA(x))_{h(i)} = (I'A'(x))_i \quad \forall x \in \Sigma^*$$

when

$$(I'A'(x))_i F_i \neq 0 \quad .$$

Hence

$$\sum_{i=1}^{n} (IA(x))_{h(i)} F_{h(i)} = \sum_{i=1}^{n} (I'A'(x))_i F_i'$$

or

$$E_A(x) = E_{A'}(x)$$

which is expectation equivalence. For any finite N

$$F_{h(i)}^N = (F_i')^N$$

The fact that

$$\mu_N^A(x) = \mu_N^{A'}(x)$$

comes from inspection of Theorem 2.1. Symbolically, we have shown $A \equiv_D A' \Rightarrow A \equiv_N A'$ for any N. How close one can come to a converse to Theorem 2.2 depends on the form of the entires of F.

Lemma 2.1 (Gantmacher [11])

Given a sequence $S_O, S_1, \ldots$ of real numbers S, if one determines positive numbers

$$r_1 > 0,\ r_2 > 0, \ldots, r_n > 0$$

$$\infty > V_m > V_{m-1}, \ldots, V_1 > 0$$

such that the following equations hold

$$(*) \qquad S_p = \sum_{j=1}^{m} r_j V_j^p \quad (p = 0,1,2,\ldots)$$

then the solution to (*) is unique. We can apply the lemma to get the following partial converse.

Theorem 2.3 If machines A and A' meet the following requirements (Letting $h(i) = i$ W.L.G.)

(i) $(IA(x))_i F_i = 0$ iff $(I'A'(x))_i F_i' = 0 \quad i = 1,2,\ldots,n.$

(ii) All states in a given machine have distinct output symbols

(iii) $E_A(x) = E_{A'}(x) \quad \forall x \in \Sigma^*$

$\mu_i^A(x) = \mu_i^{A'}(x) \quad i = 2,3,\ldots$

Then A and A' are distribution equivalent.

Proof: We use Lemma 2.1

Since the central moments and expected values of output are equal for any string, the moments of $O_A^*(x)$ and $O_{A'}^*(x)$ about zero are equal for any string.

$$
\begin{aligned}
s_0 &= \textstyle\sum_i[(IA(x))_i \text{ such that } F_i \neq 0] \\
s_1 &= E_A(x) = E_{A'}(x) \\
s_2 &= \overset{A}{\mu_2}(x) + E_A(x)^2 = \overset{A'}{\mu_2}(x) + E_{A'}(x)^2 \\
&\vdots
\end{aligned}
$$

We discard those components whose contribution to the moment is zero and relabel the non-zero components by the index j. Let

$$J = \{i : IA(x)_i F_i \neq 0\}$$

Because of assumption (i) we also have

$$j + \{i : I'A(x)_i' F_i' \neq 0\}$$

Hence

$$s_P = \sum_{j \in J} (IA(x))_j (F_j)^P \qquad P = 0,1,2,\ldots$$

$$= \sum_{j \in J} (I'A'(x))_j (F_j')^P \qquad P = 0,1,2,\ldots$$

By the lemma the solution is unique.

$$(IA(x))_j = (I'A'(x))_j \qquad j \in J$$

$$F_j = F_j' \qquad j \in J$$

Hence A and A' are distribution equivalent.

Example 2.1

The condition (ii) of theorem 2.3 is a necessary condition as shown by the following:

$$IA(x) = (.5,\ .3,\ .2) \qquad F = F' = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

$$I'A'(x) = (.5,\ .4,\ .1)$$

$$E_A(x) = IA(x)F = .5$$

$$E_{A'}(x) = I'A'(x)F' = .5$$

Since A and A' are Rabin automata, by Corollary 2.3

$$\overset{A}{\mu_i}(x) = \overset{A}{\mu_i}(x) \quad i = 2,3,\ldots$$

However, A and A' have different distributions over states for the string x.

2.6 THE N-REDUCTION RELATION

<u>Definition 2.5</u>: The <u>N-reduction relation</u> $\underline{R_F^N}$: xR_F^Ny if for all I in S

$$[E_A(x) = E_A(y) \quad \text{and} \quad \mu_i^A(x) = \mu_i^A(y) \quad i = 2,3,\ldots,N]$$

$$\Rightarrow [E_A(xz) = E_A(yz) \quad \text{and} \quad \mu_i^A(xz) = \mu_i^A(yz) \quad \forall z \in \Sigma^*, \quad i = 2,3,\ldots,N]$$

The relation R_F^N is a congruence relation and $R_F = R_F^1$. Elements in the same congruence class of R_F^N have expectations and the first N-1 central moments equal. Hence the machine Σ^* mod R_F^N can have random devices attached to the states (which are $R_F^N[x]$) such that the first N-1 central moments and expectation of each device is the same as the congruence class represented by the state. The resulting machine has deterministic switching and random output functions and is equivalent by $\equiv_N$ to the probabilistic machine defining R_F^N.

<u>Theorem 2.4</u>

The N-reduction relation is non-trivial iff there are strings x and y:

$$A(x) = A(y) + \begin{vmatrix} h_1 \\ \vdots \\ h_n \end{vmatrix} \text{ where } < \{h_1,\ldots,h_n\} > \subset \bigcap_{i=1}^{N} \text{Kern.}(F^i)$$

$$\text{and } < \{h_1,\ldots,h_n\} > A(\sigma) \subset \bigcap_{i=1}^{N} \text{Kern.}(F^i)$$

Proof: Suppose that R_F^N is non-trivial.

$$E_A(x) = E_A(y) \qquad x \neq y$$

$$\Leftrightarrow IA(x)F = IA(y)F \quad \forall I \in S$$

$$\Leftrightarrow A(x) = A(y) + \begin{vmatrix} r_1 \\ \vdots \\ r_n \end{vmatrix}$$

$$r_i \in \text{Kern.}(F)$$

$$i = 1,2,\ldots n \quad .$$

$$\mu_A^2(x) = IA(x)(F^2) - E_A(x)^2$$

$$= IA(y)(F^2) - E_A(y)^2$$

$$IA(x)(F)^2 = IA(y)(F^2) \quad \forall I \in S$$

$$\Leftrightarrow A(x) = A(y) + \begin{vmatrix} r \\ \vdots \\ r_n \end{vmatrix} \quad r_i \in \text{Kern.}(F^2)$$

For any i, $\mu_i^A(x)$ can be written as a recursive function of $IA(x)(F^i)$ and smaller powers of F, i.e.,

$$\mu_i^A(x) = IA(x)(F^i) + \sum_{k=1}^{i-1} (-1)^k \binom{i}{k} IA(x)(F^{i-k}) E_A(x)^k + (-1)^i E_A(x)^i$$

Hence by induction we assume

$$IA(x)(F^k) = IA(y)(F^k) \quad k = 1,2,\ldots i-1; \quad \forall I$$

Hence

$$\mu_i^A(x) = IA(x)(F^i) + \beta$$

$$\mu_i^A(y) = IA(y)(F^i) + \beta$$

$$\mu_i^A(x) = \mu_i^A(y) \Longleftrightarrow IA(x)(F^i) = IA(y)(F^i) \quad \forall i$$

$$\Longleftrightarrow A(x) = A(y) + \begin{vmatrix} r_1 \\ \vdots \\ r_n \end{vmatrix} \text{ where } r_i \in \text{Kern.}(F^i)$$

The rest of the proof is analogous to Theorem 1.5. Q.E.D.

If we substitute R_F^N for R_F and $\bigcap_{i=1}^{N}$ Kern.(F^i) for Kern.(F) the proofs of Theorems 1.4, 1.6, 1.8, and 1.9 go through exactly as before and we state the dual theorems which are obtained.

Theorem 1.4D

If R_F^N has finite rank r there exists a partition $\pi = (\pi_1,\ldots,\pi_r)$ on $V(A)$ and an integer valued function $y(i,m)$ such that $\pi_i A(\sigma) = \pi_{y(i,\sigma)}$ $i = 1,2,\ldots,r$ $\sigma \in \Sigma$.

Theorem 1.6D

Let $U = < \bigcup_{x\in\Sigma^*} \{A(x)_i - A(y)_i\} \quad i = 1,2,\ldots,n >$ for $(x,y) \in R_F^N$ then for any $z \in \Sigma^*$

$$UA(z) \subset \bigcap_{i=1}^{N} \text{Kern.}(F^i) \Longleftrightarrow \text{there exists } V$$

a subspace of R^n such that for any $i \in \Sigma$

(i) $UA(i) \subset V$

(ii) $VA(i) = V \subset \bigcap_{i=1}^{N} \text{Kern.}(F^i)$

Theorem 1.8D

R_F^N is non-trivial $\Longleftrightarrow$ $(\exists V)$ a subspace R^n such that

$$\text{(i)}\quad V \subset \bigcap_{i=1}^{N} \text{Kern.}(F^i)$$

$$\text{(ii)}\quad V \text{ is invariant under } \{A(i)\}_{i \in \Sigma}$$

$$\text{(iii)}\quad A(x) = A(y) + H \text{ where } H_i \in V$$

$$\text{some } H_i \neq 0.$$

Example 2.2

We extend Example 1.3 to illustrate theorems 1.4D and 1.8D.

$$< \{(0,\ 0,\ p,\ 0,\ -p,\ 0\} > \subset \bigcap_{n=1}^{N} \text{Kern} \begin{pmatrix} 10^n \\ 5^n \\ 1^n \\ 2^n \\ 1^n \\ 2^n \end{pmatrix}$$

for any finite N.

Hence we can replace the output from any state with a random device with the same first N central moments as the probabilistic sequential machine. By way of illustration, we compute the variances. Note that here the classes of R_F are also the classes of R_F^N.

$$\mu_2^A(\Lambda) = (8/10,\ 1/10,\ 1/10,\ 0,\ 0,\ 0) \begin{pmatrix} 100 \\ 25 \\ 1 \\ 4 \\ 1 \\ 4 \end{pmatrix} - (8.6)^2$$

$$= 8.84$$

Likewise, we get

$$\overset{A}{\mu_2}(0) = 1.44$$

$$\overset{A}{\mu_2}(1) = .09$$

$$\overset{A}{\mu_2}(01) = .09$$

$$\overset{A}{\mu_2}(10) = .09$$

$$\overset{A}{\mu_2}(11) = (0,\ 0,\ 9/40,\ 0,\ 31/40,\ 0) \begin{pmatrix} 100 \\ 25 \\ 1 \\ 4 \\ 1 \\ 4 \end{pmatrix} - (1.0)^2$$

$$= 0.0$$

$$\overset{A}{\mu_2}(010) = (0,\ 0,\ 21/320,\ 0,\ 11/320,\ 72/80) \begin{pmatrix} 100 \\ 25 \\ 1 \\ 4 \\ 1 \\ 4 \end{pmatrix} - 3.61$$

$$= 0.0$$

$$\overset{A}{\mu_2}(0101) = (72/80,\ 0,\ 53/1280,\ 0,\ 75/1280,\ 0) \begin{pmatrix} 100 \\ 25 \\ 1 \\ 4 \\ 1 \\ 4 \end{pmatrix} - (9.1)^2$$

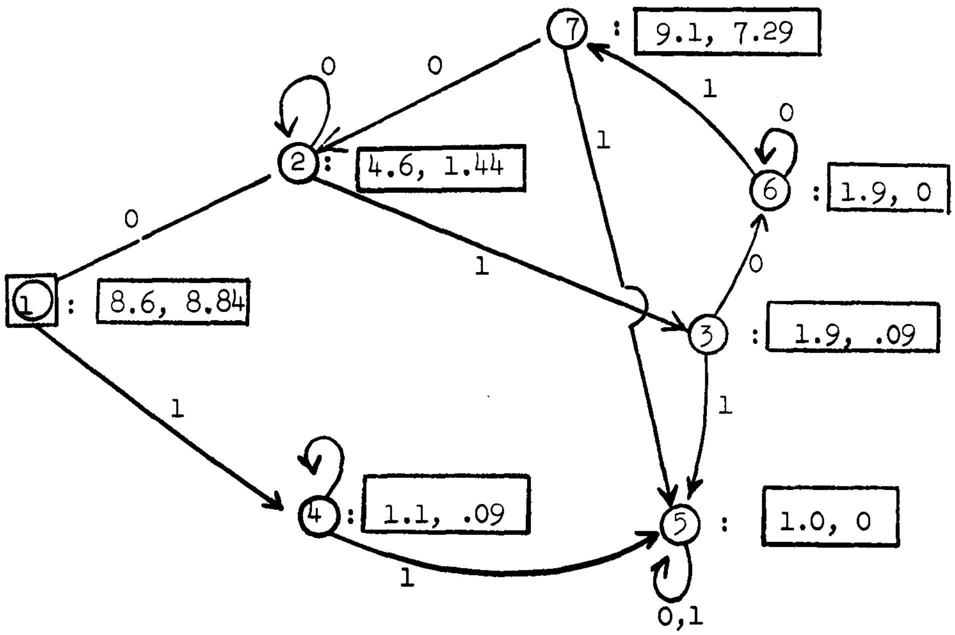
$$= 7.29$$

Hence a machine A' which has the same expectation and variance for each string can be constructed with deterministic switching and random output devices symbolized by

$$S : \boxed{e,N}$$

attached to states S which supply random numbers with mean e and variance N.

The machine A' is then just the machine of example 1.3 with the outputs connected to devices such as the above.



where $\boxed{1}$ is the initial state of A'.

Fig. 2.1 Machine A' which has the same expectation and variance for all strings as probabilistic machine A of Example 1.3.

Example 2.3. Probabilistic sequential machines A and A' such that

$$E_A(x) = E_{A'}(x) \quad .$$

and

$$\mu_i^A(x) = \mu_i^{A'}(x) \quad \forall x \in \Sigma^* \quad i = 2,3,\ldots$$

$$A(0) = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/4 & 1/4 \\ 1/4 & 0 & 3/4 \end{pmatrix} \qquad A(1) = \begin{pmatrix} 3/5 & 1/5 & 1/5 \\ 1/5 & 4/5 & 0 \\ 4/5 & 1/5 & 0 \end{pmatrix}$$

$$A'(0) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1/4 & 3/4 \\ 0 & 0 & 1 \end{pmatrix} \qquad A'(1) = \begin{pmatrix} 4/5 & 1/5 & 0 \\ 0 & 4/5 & 1/5 \\ 4/5 & 1/5 & 0 \end{pmatrix}$$

For both machines

$$F = \begin{pmatrix} F_1 \\ F_2 \\ F_1 \end{pmatrix} \text{ for } F_1,\ F_2 \text{ arbitrary real numbers} \quad .$$

Note that R_F^N is non-trivial since there is an invariant subspace

$$U = < \{(1, 0, -1)\} > \ .$$

such that

$$[A(0)-A'(0)]_j \in U$$

$$[A(1)-A'(1)]_j \in U \quad j = 1,2,\ldots n.$$

Theorem 1.9D

R_F^N is non-trivial$\Longleftrightarrow$the symbol matrices $A(i):i \in \Sigma$ be reducible for the same change of basis (f)r V) i.e. $\exists$ a linear transformation W from the state basis S to a basis for V such that

$$W^{-1}A(i)W = \begin{vmatrix} \overbrace{A_1^i}^{\text{basis for V}} & 0 \\ A_2^i & A_3^i \end{vmatrix}$$

where 0 denotes a block of all zeros the same size for all symbols i and $V \subset \bigcap_{i=1}^{N} \text{Kern.}(F^i)$.

3. THE NOTION OF INDISTINGUISHABILITY AS A CRITERION OF BEHAVIORAL EQUIVALENCE

If probabilistic sequential machines A and A' are behaviorally equivalent in an intuitive sense, taking into consideration how machines are built and repaired, one would expect them to be interchangeable as a submachine of any larger machine. Indistinguishability of two machines in any machine in which they can be plugged into is a strong criterion, the ramifications of which we shall investigate. The following example [9] illustrates how the notion of equivalence through accepting the same set of tapes, $\equiv_T$, fails to meet this indistinguishability requirement.

3.1 EXAMPLE OF TWO DISTRIBUTION EQUIVALENT MACHINES WHICH PERFORM DIFFERENTLY AS COMPONENTS OF A MACHINE

$$A_1 = < I_1, A_1(0), A_1(1), F_1 >$$

where

$$A_1(0) = A_1(1) = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$F_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \qquad I_1 = (1, 0, 0, 0, 0)$$

$$A_2 = (I_2, A_2(0), A_2(1), F_2)$$

where $I_2 = I_1$, $F_2 = F_1$

$$A_2(0) = A_2(1) = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that machines A_1 and A_2 happen to be independent of the input as $A_1(0) = A_1(1)$ and $A_2(0) = A_2(1)$ and hence are both markov processes.

TABLE 3.1

COMPARISON OF MACHINES A_1 AND A_2

x	$E_{A_1}(x)$	$I_1A_1(x)$	$E_{A_2}(x)$	$I_2A_2(x)$
Λ	0	(1, 0, 0, 0, 0)	0	(1, 0, 0, 0, 0)
0 or 1	1/2	(0, 1/2, 1/2, 0, 0)	1/2	(0, 1/2, 1/2, 0, 0)
00, 01, 10 or 11	1/2	(0, 0, 0, 1/2, 1/2)	1/2	(0, 0, 0, 1/2, 1/2)
all x: $\ell g(x) \geqq 3$	0	(0, 0, 0, 0, 1)	0	(0, 0, 0, 0, 1)

From the above table we see that A_1 and A_2 are distribution equivalent as well as expectation equivalent. We later will show the existence of a machine which behaves differently with A_1 and A_2 as submachines despite the fact that the state behaviors of A_1 and A_2 are Markov processes.

Definition 3.1. A → B denotes the machine obtained from plugging the outputs of A into the inputs of B, subject to the provision that the inputs of B be compatible with the outputs of A.

Definition 3.2. A and A' are tape equivalent machines, written $A \equiv_T A'$ if for some specified λ_1 and λ_2

$$T(A,\lambda_1) = T(A',\lambda_2) \quad .$$

Definition 3.3. A and A' are tape indistinguishable for a class $\mathcal{C}$ of machines if

$$T(A\to C\ ,\lambda) = T(A'\to C,\lambda)$$

for all λ and $C \in \mathcal{C}$.

We may sometimes let $\mathcal{C}$ be a larger class than finite deterministic or probabilistic automata.

Theorem 3.1

If probabilistic sequential machines A and A' are distribution equivalent they are not necessarily tape-indistinguishable for the class of finite deterministic automata.

Proof (by example): Let C be a finite deterministic machine which accepts 01, 10 with probability 1 and all other types with probability 0. We tabulate the expectation of $A_1 \to C$ and $A_2 \to C$ in Table 3.2.

TABLE 3.2

EXPECTATION OF $A_1 \to C$ AND $A_2 \to C$

x	$E_{A_1\to C}\ (x)$	$E_{\ A_2\to C}\ (x)$
00	0	1/4
01	1/2	1/4
10	1/2	1/4
11	0	1/4

Hence $T(A_1 \to C,\lambda) \neq T(A_2 \to C,\lambda)$ for any $\lambda \in (1/2, 0)$. The reason for this difference is because the conditional probabilities of output random variables differ for A_1 and A_2. For example,

$$\text{Prob. } \{0^*_{A_1}(01) = 1\} = 1 \quad \text{given} \quad 0^*_{A_2}(1) = 0 \ .$$

While

$$\text{Prob. } \{0^*_{A_2}(01) = 1\} = 1/2 \text{ given} \quad 0^*_{A_2}(1) = 0 \ .$$

Theorem 3.2

For probabilistic sequential machines A and A' if for all finite deterministic machines C and any cutpoint λ.

$$T(A \to C,\lambda) = T(A' \to C,\lambda)$$
$$\Longrightarrow A \equiv_E A'$$

Proof: Suppose $E_A(x) \neq E_{A'}(x)$ for some tape x of length k. Without loss of generality pick $E_A(x) > E_{A'}(x)$. Let λ_c be a rational such that $E_A(x) > \lambda_c > E_{A'}(x)$. Let C be a deterministic machine which beginning at time k computes the number $i_k - \lambda_c$ where i_k is the input at time k. Since λ_c is rational C needs only a finite number of states. C accepts the string x ff $i_k - \lambda_c \geq 0$, which can be done in a finite number of steps.

$$x \in T(B \to C,\lambda_c) \text{ iff } E_{B \to C}(x) \geq \lambda_c$$

but since C is deterministic

$$x \in T(B \to C,\lambda_c) \text{ iff } E_B(x) > \lambda_c$$

hence let B = A and B = A':

$$x \in T(A \to C,\lambda_c) \quad \text{and} \quad x \in T(A' \to C,\lambda_c)$$

so

$$T(A \to C,\lambda_c) \neq T(A' \to C,\lambda_c)$$

By logical equivalence we have shown for the class C of finite deterministic machines

$$(\lambda)(C)[T(A\to C,\lambda) = T(A'\to C,\lambda)] \Rightarrow (x)[E_A(x) = E_{A'}(x)]$$

Q.E.D.

By the example presented in Theorem 3.1 we know the converse is not true.

3.2 A MORE SATISFACTORY TECHNICAL NOTION OF INDISTINGUISHABILITY

The example at the beginning of this section shows that notions of machine equivalence such as $\equiv_T$ equivalence and even distribution equivalence, $\equiv_D$, break down under composition of machines.

In order to get a more satisfactory definition of behavioral equivalence, the conditional probability structure of probabilistic sequential machines will be explored. A stronger concept of equivalence, called indistinguishability, based upon equality for the two machines of the probabilities of all possible output strings given all possible input strings will be formulated. Following the development of Carlyle [6], a bound will be found for the length of strings needed for deciding whether two machines are indistinguishable.

In what follows it is assumed that Σ^* contains a string of one symbol Λ so that $A(\Lambda) = E_n$ the matrix identity.

Definition 3.4. The conditional probability for a sequence of outputs $y = y_1y_2\ldots y_m$ given a string of inputs $x = \sigma_1\ldots\sigma_m$ starting from an initial distribution $\Pi = (\Pi_1,\Pi_2,\ldots,\Pi_n)$ in a machine A will be written

$$P_{\Pi}^{A}(y/x)$$

or if the machine involved is clear from context, just $P_\Pi(y/x)$.

We note that the symbols of the output alphabet are real numbers which occur as components of the output column vector F, i.e. the output alphabet Y can be written

$$Y = \bigcup_{i=1}^{n} \{F_i\} \quad .$$

Definition 3.5. The probability of a sequence of transitions $S_{i_1} \to S_{i_2} \to \ldots \to S_{i_j}$ with output sequence y because of input sequence x will be written

$$P_{S_{i_1}} \to \ldots \to S_{i_j}(y/x) \quad .$$

Definition 3.6. The conditional probability transition matrix $A(y_i/\sigma)$ is formed from $A(\sigma)$ by zeroing out all columns except those corresponding to states with output y_i. More formally,

Let

$$J_{y_i} = \{j : F_j = y_i\} \quad y_i \in Y$$

and let Q^{y_i} be the matrix with $[Q^{y_i}]_{j,j} = 1$ for $j \in J_{y_i}$ and $[Q^{y_i}]_{k,j} = 0$ otherwise. Then $A(y_i/\sigma) = A(\sigma)Q^{y_i}$ $y_i \in Y$, $\sigma \in \Sigma$. Note that $[A(y_k/\sigma)]_{i,j}$ is is just $P_{S_i \to S_j}(y_k/\sigma)$.

Remark 3.6: Let $y \in Y^*$, $x \in \Sigma^*$, $y_i \in Y$, $\sigma \in \Sigma$ such that $\ell g(y) = \ell g(x)$.

Then

$$A(yy_i/x\sigma) = A(y/x)A(y_i/\sigma) \quad .$$

By definition $[A(yy_i/x\sigma)]_{l,m}$ is $P_{S_\ell \to S_m}(yy_i/x\sigma)$

For any state S_k

$$P_{S_\ell \to S_m}(yy_i/x\sigma) = P_{S_\ell \to S_k}(y/x) P_{S_k \to S_m}(y_i/\sigma)$$

since transitions to different states S_k are mutually exclusive events.

$$P_{S_\ell \to S_m}(yy_i/x\sigma) = \sum_{k=1}^{n} P_{S_\ell \to S_k}(y/x) P_{S_k \to S_m}(y_i/\sigma)$$

using the definitions again

$$[A(yy_i/x\sigma)]_{\ell,m} = \sum_{k=1}^{n} [A(y/x)]_{\ell,k}[A(y_i/\sigma)]_{k,m}$$

or in matrix form

$$A(yy_i/x\sigma) = A(y/x)A(y_i/\sigma) \quad .$$

Hence the conditional probability transition matrices for strings can be generated by the conditional probability transition matrices for symbols as was the case for the transition matrices A(x).

Remark 3.7: Given initial distribution over states Π, the probability of getting output string y from input string x is just

$$P_{\Pi}^{A}(y/x) = \sum_{j=1}^{n} \sum_{i=1}^{n} \Pi_i [A(y/x)]_{i,j}$$

with $S = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ we can write

$$P_{\Pi}^{A}(y/x) = \Pi A(y/x) S \quad .$$

Remark 3.8: We note the following identity

$$P_{II}^{A}(y/x) = \sum_{y_i \epsilon Y} P_{II}^{A}(yy_i/x\sigma) \text{ for all } \sigma \epsilon \Sigma$$

since

$$\sum_{y_i \epsilon Y} P_{II}^{A}(yy\ /x\sigma) = \sum_{y_i \epsilon Y} IIA(y/x)A(y_i/\sigma)S$$

$$= IIA(y/x) \sum_{y_i \epsilon Y} A(y_i/\sigma)S = IIA(y/x)A(\sigma)S \quad .$$

But for any n x n stochastic row matrix C

$$CS = S$$

Hence

$$IIA(y/x)A(\sigma)S = IIA(y/x)S = P_{II}^{A}(y/x) \quad .$$

Definition 3.7. The terminal distribution $II^*(y/x)$ for a sequence of outputs y given inputs x

$$II^*(y/x) = \frac{IIA(y/x)}{IIA(y/x)S} \ .$$

The i'th component of $II^*(y/x)$ is the probability of being in state i after input string x has occurred and output string y has been observed.

The following identity holds whenever $P_{II}^{A}(y/x) > 0$.

$$P_{II}^{A}(yy_i/x\sigma) = P_{II}^{A}(y/x)P_{II^*(y/x)}^{A}(y_i/\sigma) \qquad y_i \epsilon y, \quad \sigma \epsilon \Sigma$$

Definition 3.8: Machines A and A' are indistinguishable written $A \equiv_I A'$ if

$$P_{II}^{A}(y/x) = P_{II'}^{A'}(y/x) \quad \forall x \epsilon \Sigma^*, \quad \forall y \epsilon Y^*$$

Hence our concept of indistinguishability for machines depends on observable identity when both machines are started from their initial state distributions.

Definition 3.9: Machines A and A' are k-indistinguishable if

$$P^{A}_{II}(y/x) = P^{A'}_{II'}(y/x) \qquad x \in (\Sigma)^{m}, \qquad y \in (Y)^{m} \qquad \text{for} \qquad m = 0,1,\ldots,k.$$

Definition 3.10: In a machine A, two initial state distributions II and λ are indistinguishable if

$$P^{A}_{II}(y/x) = P^{A}_{\lambda}(y/x) \qquad \forall y \in Y^{*}, \qquad \forall x \in \Sigma^{*} \quad .$$

Definition 3.11: In a machine A, two initial state distributions II and λ are k-indistinguishable if

$$P^{A}_{II}(y/x) = P^{A}_{\lambda}(y/x)$$

∀x such that

$$\ell g(x) \leq k,$$

∀y such that

$$\ell g(y) = \ell g(x) \quad .$$

Checking whether the indistinguishability definition (3.8) for machines or for initial distributions (3.10) holds using only the definitions involves calculation of an unbounded sequence of conditional probabilities. In the next section is shown a bound for the length of strings whose probabilities need to be calculated. As in the deterministic machine case, if n is the number of states, then only strings of length n-2 or less need be considered in establishing indistinguishability.

3.3 THE RELATIONSHIP BETWEEN THE INTUITIVE AND TECHNICAL CONCEPTS OF INDISTINGUISHABILITY

We have yet to relate the intuitive notion of indistinguishability to the technical definition 3.8. In the next theorem will be shown that two machines indistinguishable in the technical sense are indeed indistinguishable when plugged into C, any finite state probabilistic or deterministic machine. Since C has a finite number of states, it is assumed that finite strings of $Z = C(Y^*)$, the random variable taking on values of strings of outputs of C given strings of inputs frɑm the random variable Y, depend only on finite strings Y^*.

Theorem 3.4

Let C* be the class of finite state probabilistic and deterministic sequential machines. For any C ϵ C*

$$P_{II}^{A\to C}(Z = C(Y_A^*)/x) = P_{II'}^{A'\to C}(Z = C(Y_{A'}^*)/x)$$

if

$$P_{II}^{A}(y_A/x) = P_{II'}^{A'}(y_{A'}/x)$$

for Y_A and $Y_{A'}$ having the same range $y_A \in Y^*$

Proof: For any fixed value y_A of the output string random variable of A, Y_A

$$P_{II}^{A\to C}(z = C(y_A)/x) = P_{II}^{A}(y_A/x)P^C(z = C(y_A)/y_A)$$

since the occurence of different y_A are disjoint events, for all $y_A \in Y^*$: $\ell g(y_A) = \ell g(x)$.

$$P_{II}^{A\to C}(z = C(y_A)/x) = \sum_{y_A \epsilon (Y)^{\ell g(x)}} P_{II}^{A}(y_A/x)\cdot P^{C}(z = C(y_A)/y_A)$$

Likewise

$$P_{II}^{A'\rightarrow C}(z' = C(y_{A'})/x) = \sum_{y_{A'}\epsilon(Y)^{\ell g(x)}} P_{II'}(y_{A'}/x)P^{C}(z' = C(y_{A'})/y_{A'})$$

But since Z and A' and Y_A and $Y_{A'}$ range over the same sets respectively, and the indistinguishability of A and A', i.e.

$$P_{II}^{A}(Y_A = y_A/x) = P_{II'}^{A'}(Y_{A'} = y_{A'}/x)$$

we get

$$P_{II}^{A\rightarrow C}(z = C(Y_A)/x) = P_{II'}^{A'\rightarrow C}(z' = C(Y_{A'})/x)$$

which means A→C and A'→C are indistinguishable. Q.E.D.

Since the machine C might ignore its inputs, it is clear that the converse to Theorem 3.4 does not hold.

Hence the criterion of indistinguishability as a submachine has lead us to the technical definition 3.8 as a kind of behavior equivalence.

4. FINITE COMPLETE SETS OF INVARIANTS FOR THE BEHAVIORAL EQUIVALENCES $\equiv_E$, $\equiv_N$, AND $\equiv_I$ AND THE REDUCTION CONGRUENCE RELATIONS R_F AND R_F^N

The results of the previous sections involve relations defined over all finite strings of the input alphabet. In this section are found bounds for the length of strings necessary to consider in order to decide whether two elements of the domains of the relations are in the same class.

Definition 4.1. A set of functions $f_1,\ldots,f_m$ is a *complete set of invariants* for the relation R if for all x and y in the domain of R

$$xRy \Longleftrightarrow f_i(x) = f_i(y) \quad i = 1,\ldots,m$$

We now show sets of functions which are invariants for the above relations. A set of functions which are invariant over R_F and R_F^N are:

$$\begin{array}{lcl} f_{(A,1,z)}(x) & = & E_A(xz) \\ & \vdots & \\ f_{(A,N,z)}(x) & = & \mu_N^A(xz) \end{array} \quad \text{for all } z\text{: } \ell g(z) \leq i, \quad \text{for all } I \in S$$

While for the relation $\equiv_I$, the set of functions below is a set of invariants:

$$g_{(x,y)}(A) = P_{II}^A(y/x) \quad \text{for all x and y: } \ell g(x) = \ell g(y) \leq i$$

Likewise the set

$$h_{(x,1)}(A) = E_A(x) \quad \text{for all x: } \ell g(x) \leq i$$

$$h_{(x,r)}(A) = \mu_r^A(x) \quad \text{for } r = 2,\ldots,N$$

is a set of invariants for the relations $\equiv_E$ and $\equiv_N$.

It is clear that for an unbounded i, the above are complete sets of invariants. However, in what follows a finite value of i will be found for each of these cases. In the case of $\equiv_E$ the bound will be the same as the well known Moore bound for deterministic automata but in the case of $\equiv_N$ it will be lower for most machines. The main tool used in finding the various values of i is the following simple lemma.

4.1 THE FUNDAMENTAL LEMMA

Lemma 4.1

Given an n-dimensional vector space V, a finite set $T = \{T_i\}$ where each $T_i \in V \times V$ is a linear transformation on V and some finite set of vectors $V_o \subset V$ such that $\dim \langle V_o \rangle = r \geq 1$.

Define

$$\begin{aligned} M_o &= V_o \\ M_1 &= \{v_o \cdot T_i : T_i \in T,\ v_o \in V_o\} \\ &\vdots \\ M_k &= \{v_o \cdot T_{i_1} \ldots T_{i_k} : T_{i_1}, \ldots, T_{i_k} \in T,\ v_o \in V_o\} \end{aligned}$$

and let

$$L_i = \left\langle \bigcup_{j=0}^{i} M_j \right\rangle$$

Then there exists an integer J(T) such that for any $v_o \in V_o$: $v_o \neq 0$

(i) $L_{J(T)} = L_{J(T)+1}$

(ii) $L_{k-1} \overset{\subset}{\neq} L_k$ for $k \leq J(T)$

(iii) $J(T) \leq n-r$

Proof: $L_o \subset L_1 \subset \ldots \subset L_i \subset \ldots \subset L_k$ as a consequence of the definition. The sequence $\{\dim L_j\}_{j=0}^{\infty}$ is bounded above by n, the dimension of V. Call J(T) the smallest index k such that $L_{k+1} = L_k$. Showing that the sequence $\{\dim L_j\}_{j=0}^{J(T)}$ is strictly increasing requires that for all $j+1 < J(T)$

$$L_{j+1} \neq L_{j+2} \Longrightarrow L_{j+1} \neq L_j$$

which is logically equivalent to

$$L_{j+1} = L_j \Longrightarrow L_{j+2} = L_{j+1}$$

Hence it is sufficient to show

$$L_{j+1} \quad L_j \Longrightarrow L_{j+2} \subset L_{j+1}$$

Assume

$$L_{j+1} \subset L_j$$

W.L.G. pick

$$V = V_0 \cdot T_{i_1} \ldots T_{i_{j+2}} \in L_{j+2} = (v_0 \cdot T_{i_1} \ldots T_{i_{j+1}}) \cdot T_{i_{j+2}}$$

But

$$w = v_0 \cdot T_{i_1} \ldots T_{i_{j+1}} \in L_{j+1}$$

So there is a finite set of indices $I = \{i\}$ of a spanning set U for L_j

$$U = \{v_0 \cdot T_{B_1^i} \cdot T_{B_2^i} \ldots T_{B_{r_i}^i} : i \in I\}$$

such that $r_i \leq j$ and constants c_i

$$w = \sum_{i \in I} c_i (v_0 T_{B_1^i} \ldots T_{B_{r_i}^i})$$

so

$$v = w \cdot T_{i_{j+2}} = \sum c_i (v_0 T_{B_1^i} \ldots T_{B_{r_i}^i} \cdot T_{i_{j+1}}) \in L_{j+1}$$

i.e.

$$L_{j+2} \subset L_{j+1}$$

Now consider the sequence of dimensions

$$\dim L_0, \dim L_1, \ldots, \dim L_{J(T)}$$

since

$$L_k \subsetneq L_{k+1} \quad \text{for} \quad k+1 \leq J(T), \quad \dim L_k < \dim L_{k+1} \quad \text{for} \quad k+1 \leq J(T)$$

Noting that

$$\dim L_o = r, \quad \dim L_o + J(T) \leq \dim L_{J(T)} \leq n$$

which gives

$$J(T) \leq n - r \quad .$$

Q.E.D.

4.2 A BOUND FOR TESTING FOR MEMBERSHIP IN $\equiv_I$

Theorem 4.1

If A is a probabilistic sequential machine with n states, then (n-1)-indistinguishability of initial distributions π and π' is sufficient to guarantee indistinguishability of initial distributions π and π'.

Proof: Using lemma 4.1 let

$$V_o = \{S\} = \left\{\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}\right\} \text{ and } \dim \langle V_o\rangle = 1$$

$$T = \{A(y_i/\sigma) : y_i \in Y, \quad \sigma \in \Sigma\}$$

$$V_o \cdot T_i = A(y_i/\sigma)S$$

by the lemma.

For any string $x = i_1 \ldots i_{r'}$: for r' finite, $A(y/x)S$ can be expressed as

$$A(y/x)S = \sum_{i \in I} c_i A(y_{B_1^i} \cdots y_{B_{r_i}^i} / \sigma_{j_1^i} \cdots \sigma_{j_{r_i}^i})S \qquad (*)$$

with

$$r_i \leq n - 1 \quad \text{for} \quad i \in I, \quad y_{B_k^i} \in Y, \quad \sigma_{j_k^i} \in \Sigma \quad (\text{for } k = 1, \ldots, r_i)$$

Hence for initial distributions π and π'

$$P^A_{II}(y/x) \;=\; IIA(y/x)S \;=\; \sum_{i\epsilon I} c_i IIA(y_{B_1^i}\cdots y_{B_{r_i}^i}/\sigma_{j_1^i}\cdots\sigma_{j_{r_i}^i})S$$

Let

$$y^i \;=\; y_{B_1^i}\cdots y_{B_{r_i}^i} \quad \text{and} \quad x^i \;=\; \sigma_{j_1^i}\cdots\sigma_{j_{r_i}^i}$$

$$P^A_{II}(y/x) \;=\; \sum_{i\epsilon I} c_i P^A_{II}(y^i/x^i) \quad \text{with} \quad \ell g(y^i) \;=\; \ell g(x^i) \leq n - 1$$

multiplying (*) by π' gives

$$P^A_{II'}(y/x) \;=\; \sum_{i\epsilon I} c_i P^A_{II'}(y^i/x^i)$$

By the assumption of (n-1)-indistinguishability for π and π'

$$P^A_{II}(y^i/x^i) \;=\; P^A_{II'}(y^i/x^i) \;\; \ell g(x^i) = \ell g(y^i) \leq n - 1$$

Hence

$$P^A_{II}(y/x) \;=\; P^A_{II'}(y/x) \qquad \text{Q.E.D.}$$

4.3 EQUIVALENCE OF DISTRIBUTIONS IN ONE MACHINE

Using Lemma 4.1, we can now make effective the definition of the relations R_F and R_F^N of Section 2. A bound will be obtained for the lengths of strings necessary to consider to decide whether x and y are in the same congruence class.

Definition 4.2: Distributions π and λ are equivalent for a machine A, written $\pi \underset{A}{\sim} \lambda$, if $\pi A(x)F \;=\; \lambda A(x)F \quad x \epsilon \Sigma^*$

Definition 4.3: Distributions π and λ are K-equivalent for a machine A, written $\pi \overset{K}{\underset{A}{\sim}} \lambda$, if

$$\Pi A(x) F = \lambda A(x) F \quad x \in \Sigma^*: 0 \leq \ell g(x) \leq K$$

Theorem 4.2

If A is a probabilistic sequential machine with n states and if π and λ are (n-1)-equivalent in A then $\pi \underset{A}{\sim} \lambda$.

Proof: Let x be in Σ^* and let us use Lemma 4.1 with

$V_O = \{F\}$ and $\dim \langle V_O \rangle = 1$ $T = \{A(\sigma) : \sigma \in \Sigma\}$ $V_O \cdot T_i = A(i) \cdot F$

Hence there is a finite set of vectors $A(x^i)F$ for $i \in I$ with $\ell g(x^i) \leq n - 1$ such that

$$A(x) F = \sum_{i \in I} c_i A(x^i) F \quad \ell g(x^i) \leq n - 1$$

Hence

$$\Pi A(x) F = \sum_{i \in I} c_i \Pi A(x^i) F \quad \lambda A(x) F = \sum_{i \in I} c_i \lambda A(x^i) F$$

(n-1)-equivalence gives

$$\Pi A(x^i) F = \lambda(A(x^i) F \quad i \in I$$

So

$$\Pi A(x) F = \lambda A(x) F$$ Q.E.D.

4.4 BOUNDS FOR TESTING FOR MEMBERSHIP IN $\equiv_E$ AND R_F

Definition 4.4: The abstract join of probabilistic sequential machines $A = \langle \pi, A(0), \ldots A(k-1), F \rangle$ with n states and $A' = \langle \lambda, A'(0), \ldots A'(k-1), F' \rangle$ with n' states is the abstract n+n' state machine $A^{\oplus}$ written

$$A \oplus A' = \langle \ , A^{\oplus}(0), \ldots A^{\oplus}(k-1), F^{\oplus} \rangle$$

where

$$A^{\oplus}(i) = \left(\begin{array}{c|c} A(i) & 0 \\ \hline 0 & A'(i) \end{array}\right)$$

and

$$F^{\oplus} = \left(\frac{F}{F'}\right)$$

π and λ can be embedded in the n+n' dimensional space as

$$\pi^{\oplus} = (\pi, \overbrace{0,\ldots,0}^{n' \text{ zeroes}}) \qquad \lambda^{\oplus} = (\overbrace{0,\ldots,0}^{n \text{ zeroes}}, \lambda) \quad .$$

The problem of deciding whether two machines A and A' are expectation equivalent, i.e.

$$\pi A(x) F = \lambda A'(x) F \qquad \forall x \in \Sigma^*$$

is logically equivalent to deciding when $\pi^{\oplus}$ and $\lambda^{\oplus}$ are equivalent in $A \oplus A'$, i.e. whether

$$\pi^{\oplus} \underset{A\oplus A'}{\sim} \lambda^{\oplus} \quad .$$

Hence following Caryle [6], we use Theorem 4.2 to state

Remark 4.1:

$$\pi^{\oplus} \underset{A\oplus A'}{\sim} \lambda^{\oplus} \iff \pi^{\oplus} \underset{A\oplus A'}{\overset{n+n'-1}{\sim}} \lambda^{\oplus}$$

which gives the following theorem.

Theorem 4.3

Let A and A' be probabilistic sequential machines having n and n' states respectively.

Then a necessary and sufficient condition that A and A' are expectation equivalent:

$$[\pi A(z) F = \lambda A'(z) F' \quad \forall z \in \Sigma^*] \iff [\pi A(x) F = \lambda A'(x) F' \quad \forall x\colon \ell g(x) \leq n+n'-1] \quad .$$

Theorem 4.3 makes the experimental determination of expectation equivalence possible provided the number of states of both machines are known. Furthermore, it gives a bound on the process of finding whether two strings are in the same equivalence class under the reduction relation R_F of Chapter 1. This result is summarized in the following theorem.

Theorem 4.4

Strings x and y are in the same equivalence class under the reduction relation R_F of an n state probabilistic sequential machine $A \Longleftrightarrow E_A(xz) = E_A(yz)$ for all strings z: $\ell g(z) \leq n-1$ and all $I \in S$.

Proof:

$$xR_F y \Longleftrightarrow E_A(xz) = E_A(yz) \quad \text{for all } z \in \Sigma^*, \quad \text{for all } I \in S$$

$$\Longleftrightarrow IA(x)A(z)F = IA(y)A(z)F$$

Let $\pi = IA(x)$ and $\lambda = IA(y)$

$$\Longleftrightarrow \pi A(z)F = \lambda A(z)F \quad \forall z \in \Sigma^*$$

By Theorem 4.2 and its obvious converse, we get

$$\Longleftrightarrow IA(x) \overset{n-1}{\underset{A}{\sim}} IA(y)$$

which gives the theorem.

4.5 BOUNDS FOR TESTING FOR MEMBERSHIP IN $\equiv_N$ AND R_F^N

Definition 4.5: n_F = the independence number of an n state machine A with output vector F.

$$n_F = \dim < \{(F^i) : i = 1,2,\ldots,n\} >$$

It follows from vector space arguments that

$$n_F = \# \{F_k : F_k \neq 0\}$$

The independence number is just the dimension of the space generated by powers of the components of the output vector F. For a Rabin automata $n_F = 1$ and all central moments reduce to polynominals in what we may consider the first "central moment" $E_A(x)$. In general, if the independence number is n_F, then for all x in Σ^*, the (n_F+1)'st central moment $\overset{A}{\mu}_{n_F+1}(x)$ reduces to a polynomial in the lower central moments since

$$\overset{A}{\mu}_{n_F+1}(x) = IA(x)(F^{n_F+1}) + Q(x)$$

where $Q(x)$ is a polynomial in which $IA(x)(F^i)$, $i = 1,\ldots,n_F$ occur. Hence

$$\overset{A}{\mu}_{n_F+1} = IA(x) \sum_{i=1}^{n_F} c_i(F^i) + Q(x)$$

since n_F is the dimension of the space $<(F^i) : i = 1,2,\ldots,n)>$

$$= \sum_{i=1}^{n_F} c_i IA(x)(F^i) + Q(x)$$

Theorem 4.5

Let A be a probabilistic sequential machine with output vector F and n states. Then for any $r \leq n_F$ and strings x and y in Σ^*:

$$\left\{ \begin{bmatrix} E_A(xz) = E_A(yz) \\ \overset{A}{\mu}_2(xz) = \overset{A}{\mu}_2(yz) \\ \vdots \\ \overset{A}{\mu}_r(xz) = \overset{A}{\mu}_r(yz) \end{bmatrix} \forall z \in \Sigma^* \right\} \Leftrightarrow \left\{ \begin{bmatrix} E_A(xz') = E_A(yz') \\ \overset{A}{\mu}_2(xz') = \overset{A}{\mu}_2(yz') \\ \vdots \\ \overset{A}{\mu}_r(xz') = \overset{A}{\mu}_r(yz') \end{bmatrix} \forall z' : \ell g(z') \leq n - r \right\}$$

Proof: Using Lemma 4.1 with

$$V_0 = \{F,(F^2),\ldots,(F^r)\}$$

$$\dim \langle V_o \rangle = r \leq n_F$$

$$\{T_i\} = \{A(i) : i \in \Sigma\}$$

for any $v_o \in \langle V_o \rangle$

$$v_o \cdot T_i = A(i)\, v_o = \sum_{k=1}^{r} c_k A(i)(F^k)$$

Consider any string

$$z: \ell g(z) = m' \text{ finite} \quad .$$

Then there exists a spanning set $A(x^i)\, v_o$ with $i \in I$ and constants $c_i(v_o)$ so that

$$A(z)\, v_o = \sum_{i \in I} c_i(v_o)\, A(x^i)\, v_o \; : \; \ell g(x^i) \leq n - r$$

Let v_o range over the (F^i) $i = 1,2,\ldots,r$. For any π and λ there are constant functions depending on (F^i), $c_i((F^i))$, such that

$$\pi A(z)(F^i) = \sum_{i \in I} c_i((F^i))\, \pi A(x^i)(F^i)$$

$$\lambda A(z)(F^i) = \sum_{i \in I} c_i((F^i))\, \lambda A(x^i)(F^i)$$

Hence the moments about zero from π and λ are equal if they are equal for all strings of length $\leq$ n-r. Let $\pi = IA(x)$ and $\lambda = IA(y)$. Then we have for any z and any initial distribution

$$IA(xz)(F^i) = IA(yz)(F^i) \quad i = 1,2,\ldots,r$$

holds if and only if for $i = 1,2,\ldots,r$

$$IA(xz')(F^i) = IA(yz')(F^i)$$

for all strings z' of length less than or equal to n-r. Noting by Theorem 2.1

that any central moment $\mu_m^A(x)$ is a function of $IA(x)(F),\ldots,IA(x)(F^m)$ the result is established. Q.E.D.

Corollary 4.5 (Bound for the relation R_F^N to hold)

Let A be a probabilistic sequential machine with n states and with $N \leq n_F$. Then $xR_F^N y \Longleftrightarrow$ for all strings z': $\ell g(z') \leq n-N$

$$\left\langle \begin{bmatrix} E_A(xz') & = & E_A(yz') \\ \mu_2^A(xz') & = & \mu_2^A(yz') \\ \vdots & & \\ \mu_N^A(xz') & = & \mu_N^A(yz') \end{bmatrix} \right\rangle \quad \text{for all } I \in S$$

Theorem 4.6

Let A and A' be probabilistic sequential machines having n and n' states respectively. Then for all

$$r \leq n_F + n_{F'} - \# \{\hat{y} : \hat{y} \in Y \cap Y' \text{ and } \hat{y} \neq 0\}$$

and for any initial distributions π in A and λ in A' then

$$\left\langle \begin{bmatrix} E_A(x) & = & E_{A'}(x) & \\ \mu_2^A(x) & = & \mu_2^{A'}(x) & \forall x \in \Sigma^* \\ \vdots & & & \\ \mu_r^A(x) & = & \mu_r^{A'}(x) & \end{bmatrix} \right\rangle \Longleftrightarrow \left\langle \begin{bmatrix} E_A(x') & = & E_{A'}(x') & \\ \mu_2^A(x') & = & \mu_2^{A'}(x') & \forall x' : \ell g(x') \leq n+n'-r \\ \mu_r^A(x') & = & \mu_r^{A'}(x') & \end{bmatrix} \right\rangle$$

Proof: Construct $A^{\oplus} = A \oplus A'$ and let V_o in Lemma 4.1 be

$$V_o = \{\binom{F}{F'},\ldots,\binom{F^{n+n'}}{F'^{n+n'}}\} \qquad n_F^{\oplus} = \dim \langle V_o \rangle$$

$$= \{\hat{y}:(\hat{y}\in Y \text{ or } \hat{y}\in Y') \text{ and } \hat{y} \neq 0 \text{ and } \hat{y}\notin Y\cap Y'\}$$

$$= n_F + n_{F'} - \#\{\hat{y}:\hat{y}\in Y\cap Y' \text{ and } \hat{y} \neq 0\}$$

Using Lemma 4.1 and an argument like the one in Theorem 4.4 establishes the theorem. Q.E.D.

4.6 DISCUSSION OF THE GENERALIZATION OF THE MOORE BOUND

Corollary 4.6

Let A and A' be n-state deterministic machines with two-valued output alphabet $Y = Y' = \{1,2\}$. Then A and A' are indistinguishable for all strings if they are indistinguishable for all strings of length at most 2n-2.

Proof: In Theorem 4.6 we have $n_F^{\oplus} = 2+2-2 = 2$ so that $r \leq 2$. For deterministic machines, indistinguishability reduces to $E_A(x) = E_{A'}(x)$ for all $x \in \Sigma^*$ and also

$$E_A(x) = E_{A'}(x) \Rightarrow \mu_2^A(x) = \mu_2^{A'}(x)$$

Hence the right side of Theorem 4.6 gives the result. Q.E.D.

Theorem 4.6 can be regarded as a generalization of the Moore result [7] to probabilistic machines with arbitrary rather than binary output alphabets. Note that Moore's bound is 2n-1 since he considers the initial output as part of the experiment. We consider the initial outputs when considering strings of length 1 since the symbol Λ has identity symbol matrix.

The role of the zero output symbol in Theorem 4.6 is a significant departure from Moore's deterministic results. In order to get the same result as Moore in Corollary 4.6 it was necessary to pick a two-valued output set $\{1,2\}$ rather than $\{0,1\}$ with the implicit assumption that such recoding of output symbols cannot affect indistinguishability between deterministic machines. Without the recoding, $r = 1$ and the bound is one higher than the Moore bound.

However, in the probabilistic case, a different bound for machines with a zero output symbol than those with nonzero symbols seems reasonable. A zero annihilating some probabilities in the expectation and higher moments can mask significant changes in distributions. It is clear from Theorems 1.8 and 1.8D that changes in F_1 from zero to nonzero can affect the kernel of F, perhaps to the extreme of making R_F^N of infinite rather than finite rank and preventing the construction of an N-moment equivalent finite machine with deterministic switching.

REFERENCES

[1] Braines, S. N. and Svechincky, V. B., "Matrix Structure in Simulation of Learning" I.R.E. Transactions on Information Theory Vol. IT-S, No. 5, September 1962, pp. 186-190.

[2] Howard, Ronald A., Dynamic Programming and Markov Process, M.I.T. Press and John Wiley and Sons, Inc., New York, 1960 (Chapter 3).

[3] Thrall, Robert M. and Tornheim, Leonard, Vector Spaces and Matrices, John Wiley and Sons, Inc., New York, 1957, pp. 298-300.

[4] Rabin, M. O., "Probabilistic Automata," Sequential Machines, Selected Papers, Edited by E. F. Moore, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1964, pp. 98-114.

[5] Burks, Arthur W., "Computation, Behavior and Structure in Fixed and Growing Automata," Behavioral Science, Vol. 6, N . 1, January 1961.

[6] Carlyle, J. W., "Equivalent Stochastic Sequential Machines," Institute of Engineering Research Report Series, No. 60, Issue No. 415, Electronics Research Laboratory, University of California, Berkeley, California, 1961.

[7] Moore, E. F., "Gedanken-experiments on Sequential Machines," Automata Studies, C. E. Shannon and J. McCarthy eds., Princeton University Press, 1956.

[8] Shannon, C. E. and Weaver, Warren, The Mathematical Theory of Communication, University of Illinois Press, Urbana, 1948, pp. 34.

[9] Arnold, Richard, unpublished communication.

[10] Rabin, M. O., and Scott D., "Finite Automata and Their Decision Problems," IBM Journal Res. and Dev., 3, 1959, pp. 114-125.

[11] Gantmacher, F. R., The Theory of Matrices, Vol. 2, Chalsea Publishing Co., N.Y., N.Y., 1959, pp. 236-237.

DISTRIBUTION LIST

(One copy unless otherwise noted)

Technical Library
Director Defense Res. & Eng.
Room 3C-128, The Pentagon
Washington, D.C. 20301

Defense Documentation Center 20
Cameron Station
Alexandria, Virginia 22314

Chief of Naval Research 2
Department of the Navy
Washington 25, D.C.
Attn: Code 437, Information
Systems Branch

Director, Naval Research Laboratory 6
Technical Information Officer
Washington 25, D.C.
Attention: Code 2000

Commanding Officer 10
Office of Naval Research
Navy 100, Fleet Post Office Box 39
New York, New York 09599

Commanding Officer
ONR Branch Office
207 West 24th Street
New York 11, New York

Office of Naval Research Branch
Office
495 Summer Street
Boston, Massachusetts 02110

Naval Ordnance Laboratory
White Oaks, Silver Spring 19
Maryland
Attn: Technical Library

David Taylor Model Basin
Washington,D.C. 20007
Attn: Code 042, Technical Library

Naval Electronics Laboratory
San Diego 52, California
Attn: Technical Library

Dr. Daniel Alpert, Director
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

Air Force Cambridge Research Labs
Laurence C. Hanscom Field
Bedford, Massachusetts
Attn: Research Library, CRMXL R

U. S. Naval Weapons Laboratory
Dahlgren, Virginia 22448
Attn: G. H. Gleissner, Code K4
Asst. Dir. for Computation

National Bureau of Standards
Data Processing Systems Division
Room 239, Building 10
Washington 25, D.C.
Attn: A. K. Smilow

George C. Francis
Computing Laboratory, BRL
Aberdeen Proving Ground, Maryland

Office of Naval Research
Branch Office Chicago
230 N. Michigan Avenue
Chicago, Illinois 60601

Commanding Officer
ONR Branch Office
1030 E. Green Street
Pasadena, California

Commanding Officer
ONR Branch Office
1000 Geary Street
San Francisco 9, California

DISTRIBUTION LIST (Concluded)

The University of Michigan
Department of Philosophy
Attn: Professor A. W. Burks

National Physical Laboratory
Teddington, Middlesex, England
Attn: Dr. A. M. Uttley, Supt.
Autonomics Division

Commanding Officer
Harry Diamond Laboratories
Washington, D.C. 20438
Attn: Library

Commanding Officer and Director
U. S. Naval Training Device Center
Port Washington
Long Island, New York
Attn: Technical Library

Department of the Army
Office of the Chief of Research
and Development
Pentagon, Room 3D442
Washington 25, D.C.
Attn: Mr. L. H. Geiger

National Security Agency
Fort George G. Meade, Maryland
Attn: Librarian, C-332

Lincoln Laboratory
Massachusetts Institute of Technology
Lexington 73, Massachusetts
Attn: Library

Office of Naval Research
Washington 25, D.C.
Attn: Code 432

Kenneth Krohn
6001 Dunham Springs Road
Nashville, Tennessee

Mr. Laurence J. Fogel
General Dynamics/Astronautics
Division of General Dynamics Corp.
San Diego, California

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Logic of Computers Group
The University of Michigan
Ann Arbor, Michigan 48104

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

EQUIVALENCES BETWEEN PROBABILISTIC AND DETERMINISTIC SEQUENTIAL MACHINES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(Last name, first name, initial)*

Page, Carl V.

6. REPORT DATE

April, 1965

7a. TOTAL NO. OF PAGES

67

7b. NO. OF REFS

11

8a. CONTRACT OR GRANT NO.

Nonr 1224(21)

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

03105-37-T

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Office of Naval Research
Department of the Navy
Washington 25, D.C.

13. ABSTRACT

The concept of probabilistic sequential machines (PSM), a generalization of Rabin's concept of probabilistic automata, is defined. Such diverse devices as unreliable digital computers, slot machines, and chemical cells are presented as examples of PSM. Using the examples as motivation, various kinds of equivalences between machines are discussed. The fundamental question of when a PSM is equivalent in some sense to a deterministic machine, perhaps with random devices attached to output states, is considered. Finally various tests involving finitely many random variables are devised for each of the kinds of equivalences between PSM and for reduction, if possible, to deterministic machines. One of the tests is a further generalization of the Moore bound for deterministic machines than has previously appeared in the literature. (U)

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
1. probabilistic						
2. sequential machine						
3. automata theory						
4. communications channel						
5. stochastic machines						
6. discrete stochastic processes						
7. markov processes with rewards						
8. congruence relations						
9. probabilistic automata						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
_______________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
_______________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
_______________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.